Rapport de Stage

Détection d'Anomalies

DING WENSI

August 25, 2016

1 Introduction

Ayant une base de données sur la géolocalisation des utilisateurs téléphoniques, ce projet a pour but de détecter des imsi en question : les personnes qui ont soit des comportements inhumains, soit des comportements particuliers par rapport à la plupart des gens. Certainement, il peut aussi avoir des imprécisions ou erreurs sur des données qui rendent des comportements étranges. Toutes ces catégories rentrent dans notre définition d'anomalie. Le processus de détection d'anomalies nous permet de mieux comprendre les données et enlever les bruits sur les données pour les analyses futures.

2 Modélisation

2.1 Présentation des Modèles

2.1.1 Modèle Gaussien

Le modèle Gaussien est un modèle qui se base sur des méthodes statistiques. En supposant que chaque variable suit une loi gaussienne et toutes les variables sont indépendantes, la probabilité de chaque point peut se calculer facilement. L'hypothèse est que les points normaux sont ceux qui ont une grande probabilité, et les points anormaux ont une tendance d'avoir une petite probabilité.

2.1.2 Modèle LOF

Le modèle LOF est un modèle qui se base sur l'algorithme k plus proches voisins. L'hypothèse est que les points normaux restent proches mais les anomalies sont loin de leurs voisins. Ayant tous les avantages des modèles qui détecte des anomalies directement par la densité, ce modèle est capable de trouver des anomalies locales en plus. Les anomalies locales sont une sorte d'anomalie qui ont une densité très différente de celles des points à côté, mais ses propres densités restent raisonnables. Par exemple, dans le schéma à côté, les distances entre le point o_2 et ses plus proches voisins sont à peu près à la même grandeur que la distance entre n'importe quels deux points dans le cluster C_1. Il n'y a aucun moyen de prendre le point o_2 comme une anomalie sans rendre les points dans le cluster C_1 anormaux avec la notion de densité globale. Du coup, l'idée principale est de calculer le rapport entre la densité des points proches et sa propre densité pour chaque point. Plus ce rapport est élevé, plus le point a de possibilité d'être une anomalie. Pour une présentation plus en détail sur ce modèle, des diapo sont disponibles sur le wiki d'équipe data science.

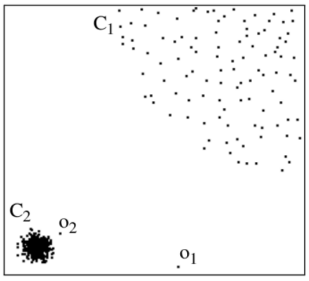
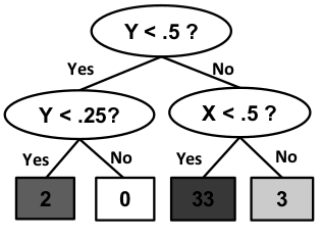


Figure 1: 2-*d* dataset DS1

2.1.3 Modèle Half Space Tree

Le modèle half space tree est une variante d'arbre de décision. Chaque nœud dans l'arbre représente des intervalles de toutes les dimension. L'initialisation est faite par la division égale en deux parties d'une des dimensions choisie au hasard. Et puis ce processus est répété plusieurs fois en descendant d'arbre. Un exemple simple est illustré à droite. L'hypothèse est que s'il y a moins de points qui rentrent dans la même feuille, il y a plus de probabilité que ces points sont anormaux. Comme la profondeur d'un arbre est limitée par la mémoire interne,



un ensemble d'arbres est utilisé, pour que notre modèle soit le plus
précis que possible. Au lieu d'estimer sur la distribution de l'ensemble de données, ce modèle propose de garder seulement la distribution d'un certain nombre de données en mémoire et d'estimer les données qui viennent à partir de ces mémoires et puis mettre ces dernières données en mémoire pour la suite. Le processus continue sans garder toute l'histoire. Avec cette méthode, nous pouvons espérer détecter les anomalies même si elles évoluent. Mais dans notre cas d'étude, nous l'utilisons dans le cadre statique.

2.2 Implémentation des Modèles

2.2.1 Présentation des données

Deux types de données sont fournis. Tout à bord, ce sont des données brutes : un tableau avec onze champs au total. Parmi ces champs, seulement cinq sont utilisés et ils sont listés au dessous.

- timestamp : un format de date correspondant au nombre de secondes depuis le 01/01/1970
- tac : code lié au type d'appareil contenant la carte SIM
- event: type d'événement (entrée ou sortie d'une cellule, etc...)
- imsi : identifiant (anonymisé) de la carte SIM de l'utilisateur
- cgi : identifiant de la cellule du téléopérateur

Un ficher supplémentaire qui donne des informations comme les longitudes et les latitudes des centres et aussi les rayons de chaque cgi est en accompagnement. Donc pour un utilisateur (imsi) donnée, nous avons toutes les lignes de données dans le tableau qui nous indiquent quel type d'événement (event) a été fait à quel moment (timestamp) dans quel région (cgi, sa longitude, et sa latitude).
Un autre type de données qui s'appelle **stop segment** est obtenu par des traitements sur les données brutes. Dans ce type de données, les événements d'un individu sont réduits à deux types : STOP et MOVE. Si cet individu reste suffisamment de temps dans une région pas grande, alors tous les événements pendant cette période sont représentés par un STOP. Entre deux STOP, c'est un MOVE. Pour chaque MOVE et STOP, nous avons aussi des informations enregistrées dans un tableau avec plusieurs champs. Certains que nous utilisons sont listés au dessous.

- imsi : identifiant (anonymisé) de la carte SIM de l'utilisateur
- tsStart : timestamp du premier événement dans l'ensemble que MOVE ou STOP contient
- tsEnd : timestamp du dernier événement dans l'ensemble que MOVE ou STOP contient
- cuidStart : identifiant de la cellule du premier événement dans l'ensemble que MOVE ou STOP contient
- cuidEnd : identifiant de la cellule du dernier événement dans l'ensemble que MOVE ou STOP contient
- speed : vitesse correspondante
- segmentType : MOVE ou STOP

2.2.2 Pré-traitement sur les données brutes

Avant de faire les calculs sur les variables, certaines mesures sont effectuées afin que le calcul des variables ne soit pas trop perturbé.

Remarques sur les données brutes :	**Propositions :**
très peu de données sur certains imsi tel que l'on risque d'être incapable de calculer certaines variables ou d'avoir des valeurs très biaisées	tester d'abord sur les imsi qui ont des données assez importantes
les valeurs disponibles pour les différents champs varient selon le type d'événement, le nombre d'entrée et le nombre de sortie n'est pas toujours cohérent	prendre seulement les événements de type "CELL_ENTER"
le nombre de jours tel que les données sont disponibles varie pour différents imsi, pas de données pendant quelques jours au milieu pour un imsi	créer les variables qui se basent sur le comportement de chaque jour au lieu de l'ensemble de jours
les entrées consécutives dans la même cellule ou des différentes cellules mais avec la même position pour un imsi	garder seulement la première entrée
différents événements se passent au même moment pour un imsi	enlever les intervalles (deux événements consécutifs) avec la durée de zéro
oscillations rapides entre deux cellules voisines	essayer de les enlever en rejetant les intervalles avec la distance courte et la vitesse grande

Table 1: Pré-traitements sur les données brutes

2.2.3 Variables Choisies

La première partie pour implémenter les modèles est de choisir des variables discriminantes pour filtrer les anomalies. Du fait que notre but est de trouver des imsi qui ont des comportements absurdes en tant qu'être humain, les variables qui sont capables de refléter la mobilité d'une personne sont préférables.

Tout à bord, on essaie de créer des variables directement à partir des données brutes. Quelques idées sont présentées ici.

noms :	**explications :**
v_max	le maximum des vitesses entre deux événements consécutifs
v_mean	la moyenne sur l'ensemble des valeurs moyennes des vitesses entre deux événements consécutifs chaque jour
n_events_mean	la moyenne du nombre d'événements chaque jour
n_events_sd	la variance du nombre d'événements chaque jour
duree_stop_mean	la moyenne des sommes de durée d'être immobile chaque jour
duree_stop_sd	la variance des sommes de durée d'être immobile chaque jour

Table 2: Variables basées sur les données brutes

Avec les fichiers de stop-segment fournis, des nouvelles variables peuvent être créées. Quelques-unes sont implémentées et testées.

noms :	**explications :**
v_max_ss	le maximum des vitesses pour tous les événements de type "MOVE"
v_mean_ss	la moyenne des valeurs moyennes des vitesse chaque jour pour tous les événements de type "MOVE"
n_stop_mean	la moyenne du nombre d'événements de type "STOP" chaque jour
n_stop_sd	la variance du nombre d'événements de type "STOP" chaque jour
duree_stop_mean_ss	la moyenne des sommes de durée des événements de type "STOP" chaque jour
duree_stop_sd_ss	la variance des sommes de durée des événements de type "STOP" chaque jour
dis_ss	la médiane des distances entre les endroits où la personne a fait le dernier stop chaque jour

Table 3: Variables basées sur les données de stop-segment

2.2.4 Quelques détails

La première étape et aussi l'étape commune pour tous les trois modèles est d'obtenir les valeurs des variables pour chaque imsi données. Et puis, pour la partie de **normalisation**, on choisit de faire une normalisation simple sur l'ensemble de variables: la vraie valeur est remplacée par la différence entre la vraie valeur et la valeur minimale sur la différence entre la plus grande et la plus petite valeur pour une variable donnée. Si certains imsi prennent la valeur **NA** sur une variable, on fait le choix de les remplacer par la valeur moyenne de cette variable. En plus, pour le modèle gaussien, la **transformation de variables** doit être mise en place pour que la distribution de chaque variable soit vraiment gaussienne.

2.3 Méthodes d'Évaluation

2.3.1 Évaluation des modèles

Pour évaluer des modèles, deux approches : matrice de confusion et courbe ROC sont les plus populaires. Comme nos trois modèles nous renvoient une sorte de score pour indiquer le niveau d'anomalie, un seuil devrait être choisi pour finalement faire la prédiction à partir des résultats obtenus. Néanmoins, la courbe ROC est capable de mesurer la qualité d'un modèle sans avoir un seuil choisi. Par contre, il faut transformer les scores dans un intervalle de zéro à un. Pour ce faire, on décide de mettre la partie avec de très grands scores directement à un et puis une transformation linéaire pour les autres.

2.3.2 Évaluation des hyperparamètres

Chaque modèle contient un certain nombre d'hyperparamètres prédéfinis qui ne sont pas à apprendre dans le modèle. Par exemple, un hyperparamètre pour tous les modèle est les variables choisies parmi toutes les possibilités. Pour trouver les meilleurs hyperparamètres, grid search est souvent utilisé. L'idée est d'évaluer le modèle avec tous les hyperparamètres possibles. Les possibilités d'hyperparamètres sont composés par toutes les combinaisons de valeurs possibles. Mais une autre méthode Random search propose de faire l'évaluation seulement sur un certain nombre des hyperparamètres pris au hasard. Avec cette méthode, nous évaluons sur beaucoup moins d'hyperparamètres, mais nous arrivons quand même à obtenir un résultat comparable par rapport à grid search.

Dans le cadre de python, deux fonctions sont demandées pour faire le grid search ou le random search. Une fonction "fit" définit le processus d'apprentissage, une fonction "score" définit le processus d'évaluation de modèle. Pour un ensemble de données en entrée et aussi le nombre de cross-validation **n** indiqué, les données seront divisées en **n** parties. Pour un hyperparamètre, la fonction "fit" apprend sur les **n-1** parties prises et puis la fonction "score" donne un score par la performance du modèle sur la partie restée. En permutant les parties prises par "fit", **n** apprentissages et évaluations seront faits. Le score final pour un hyperparamètre est la valeur moyenne sur l'ensemble de scores retournés par la fonction "score".

3 Test et Conclusion

3.1 Test et Conclusion sur les variables

Pour savoir et aussi comparer les différents effets de chaque variable, quelques tests sont mis en place. Pour une variable choisie, on fait d'abord une distribution gaussienne sur les valeurs de cette variable pour tous les imsi et on ordonne les imsi par la densité de probabilité obtenue. En examinant les top imsi avec le moins de probabilité, quelques conclusions sont listées dans le tableau suivant.

noms :	**conclusions :**
v_max	capable de sortir deux types d'anomalies: cellules mal positionnées, oscillations entre deux cellules avec une distance grande
v_max_ss	capable d'enlever les événements anormaux à cause de cellules très mal placées; pour les cellules pas trop mal positionnés ou les cellules avec rayons grands, le résultat peut varier
v_mean	écrasé par les v_max très grands, pas très performant
v_mean_ss	capable de représenter une sorte de mobilité d'une personne en moyenne
n_events_mean	capable de sortir des imsi avec le nombre d'oscillations entre plusieurs cellules très grand
n_stop_mean	capable de représenter une sorte de mobilité d'une personne, exemple: un voyageur peut faire plein de stop pendant une journée
n_events_sd	écrasé par les n_events_mean très grands, pas très performant
n_stop_sd	capable de représenter une sorte de mobilité d'une personne, exemple: un voyageur qui visite plusieurs villes peut faire plein de stop dans une journée de visite, très peu de stop dans une journée de déplacement
duree_stop_mean	pas très précis pour représenter la vraie durée de stop
duree_stop_sd	pas très précis à cause d'imprécision de duree_stop_mean
duree_stop_mean_ss	capable de représenter une sorte de mobilité d'une personne, mais cette variable a une relation linéaire avec n_stop_mean
duree_stop_sd_ss	pas de typique comportement sorti
dis_ss	capable de représenter une sorte de mobilité dans le sens global, exemple: les personnes qui bougent tout le temps avec une grande distance chaque jour

Table 4: Conclusion sur les variables

Après des analyses en détail sur chaque variable, quelques remarques sont listées dans le tableau au dessus. En prenant en compte la performance des variables possibles, les variables : v_max, v_mean_ss, n_events_mean, n_stop_mean, n_stop_sd sont choisies finalement. Bien que la variable **dis_ss** fournisse aussi une bonne mesure, elle n'est pas convenable pour notre période de test où la plupart des gens bougent beaucoup.

3.2 Test et Conclusion sur les modèles

Tous les trois modèles sont implémentés deux fois avec deux ensembles différents de variables. Le premier ensemble de variables (E1) est toutes les six variables qui se basent seulement sur les données brutes. Et le deuxième ensemble de variables (E2) est toutes les cinq variables choisies dans la partie précédente. Pour l'implémentation des trois modèles sur chaque ensemble de variables, trois fichiers de résultat sont sortis. Du fait que nous n'avons pas vraiment les étiquettes sur les imsi, dans un premier temps, on aimerait comparer les trois modèles. Ceci peut être fait en gardant un des résultats comme une référence et en faisant l'évaluation sur les

deux autres.

Premièrement, la méthode de courbe ROC est réalisée. Cette méthode est supposée de bien fonctionner même dans le cas où le résultat est super biaisé. Pour garder le résultat du modèle gaussien comme une référence, un seuil approprié est choisi pour avoir des étiquettes sur les imsi. Les AUC scores obtenus sont listés ici :

- LOF : 0.78 (E1) ; 0.82 (E2)
- HS_Tree : 0.94 (E1) ; 0.93 (E2)

Avec ce résultat, on a envie de dire que le modèle HS_Tree ressemble beaucoup au modèle gaussien. Mais avec une analyse plus en détail, on peut découvrir que les trois modèles ne sont pas si proches comme les AUC scores indiquent. La méthode de matrice de confusion est implémentée en suite. Pour que les résultats soient comparables, un nombre fixe d'anomalies pour les trois modèles est choisi. Le tableau suivant correspond au cas de 679 anomalies sur l'ensemble de 157107 imsi. Et ce sont des résultats sur le deuxième ensemble de variables.

Modèle Référent :	Gaussien	Gaussien	LOF
Modèle Testé :	LOF	HS_Tree	HS_Tree
Accuracy Score :	0.99	0.99	0.99
F1 Score :	0.23	0.22	0.11
Matrice de Confusion :	TN : 155906; FP, FN : 522; TP : 157	TN : 155897; FP, FN : 531; TP : 148	TN : 155825; FP, FN : 603; TP : 76

Table 5: Résultat matrice de confusion

Nous pouvons remarquer facilement qu'il existe des grandes différences sur les anomalies des trois modèles. Le même comportement est présenté aussi pour le premier ensemble de variables. La raison que les AUC scores sont quand même proches à un est simple. Un nombre d'anomalies pour le modèle référent (679 dans le cas présenté) est fixés. Si nous baissons un petit peu le seuil pour le modèle testé, le nombre d'anomalies augmente jusqu'à par exemple 3000. Alors, nous pouvons arriver à obtenir un très bon chiffre de true positive sans baisser trop le chiffre de true negative. Parce que les anomalies représentent un très petit ensemble (moins de 1%) sur les données entières. Par conséquent, la courbe ROC n'est pas très performante dans ce cas comme nous supposons.

Comme le but est de comparer les modèles, la similarité de Jaccard peut être un bon indicateur. Si on note le nombre d'anomalies pour tous les modèles avec la lettre **k**, alors finalement la similarité de Jaccard peut s'écrit : $\frac{TP}{TP+FN+FP} = \frac{TP}{2k-TP}$. Ce nombre est grand seulement quand TP est assez grand. Du coup, nous n'avons plus le risque d'être biaisé par le nombre de true negative.

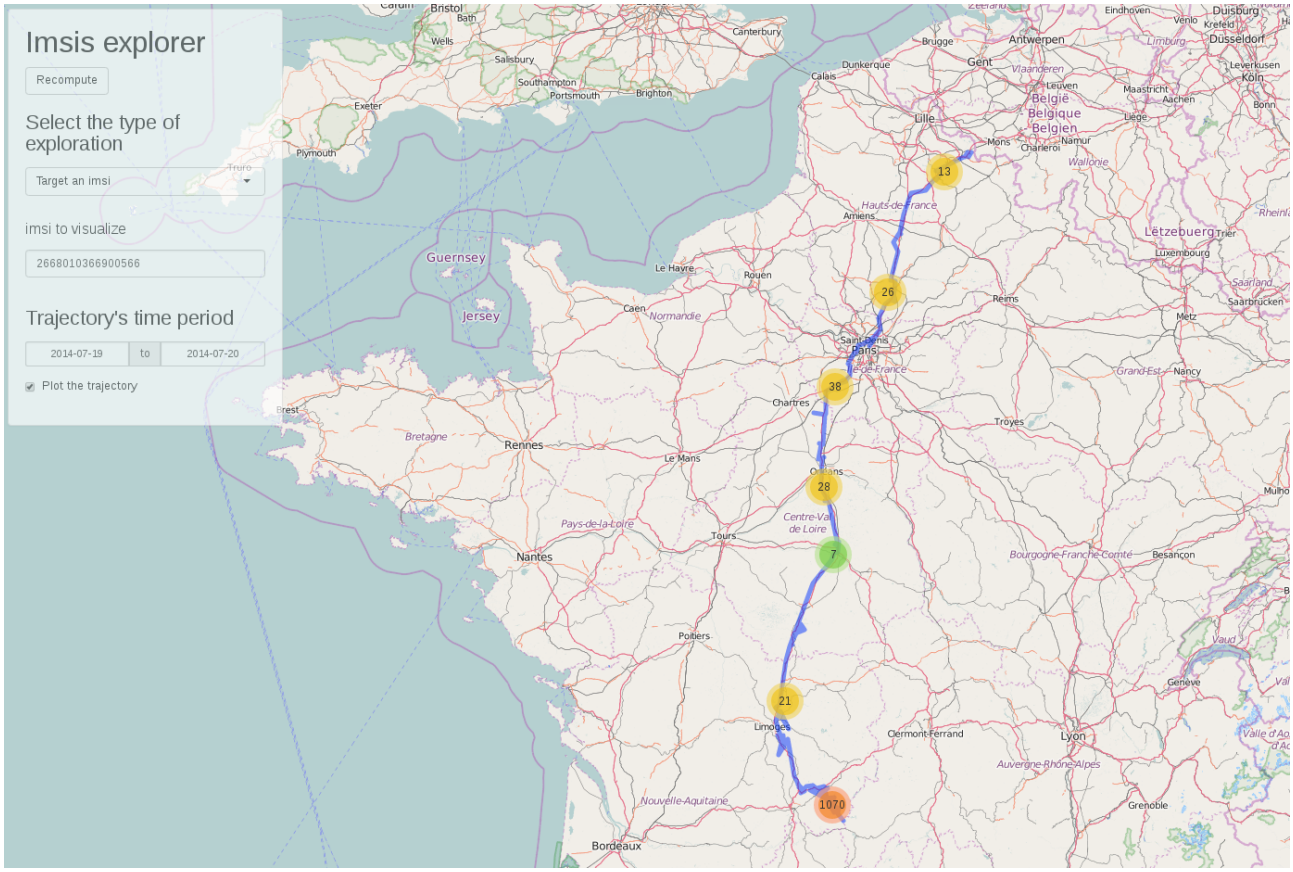
Après avoir examiné les top anomalies de tous les trois modèles, une des explications possible pour la différence entre le modèle gaussien et les deux autres est découverte. Et cet effet est plus remarquable dans les résultats sur les variables du deuxième ensemble. Donc l'explication suivante est basée sur les résultats de E2. Dans les variables choisies, les deux variables : v_max et

n_events_mean peuvent prendre des valeurs énormes par rapport à des valeurs normales. Même si une normalisation simple et linéaire est faite pour toutes les variables, le gros décalage reste toujours. Du coup, dans le cas de modèle LOF et modèle HS_Tree, ces deux variables peuvent contribuer beaucoup plus à la distance entre les points. Et en conséquence, elles deviennent plus discriminantes. Mais dans le modèle gaussien, une transformation sur les variables pour que la distribution soit gaussienne est faite. Typiquement, cette transformation est de prendre la puissance 0.1 ou 0.2 de la vraie valeur d'une variable. Alors, le gros décalage peut être beaucoup réduit et les deux variables ne sont plus discriminantes pour le modèle gaussien. Ceci peut être montré par le fait que les top anomalies de modèle gaussien sont souvent des imsi qui prennent des valeurs aberrantes sur les variables basées sur stop-segment. Au contraire, les top anomalies pour les deux autres modèles sont plutôt des imsi qui prennent des valeurs aberrantes pour les deux variables indiquées avant.

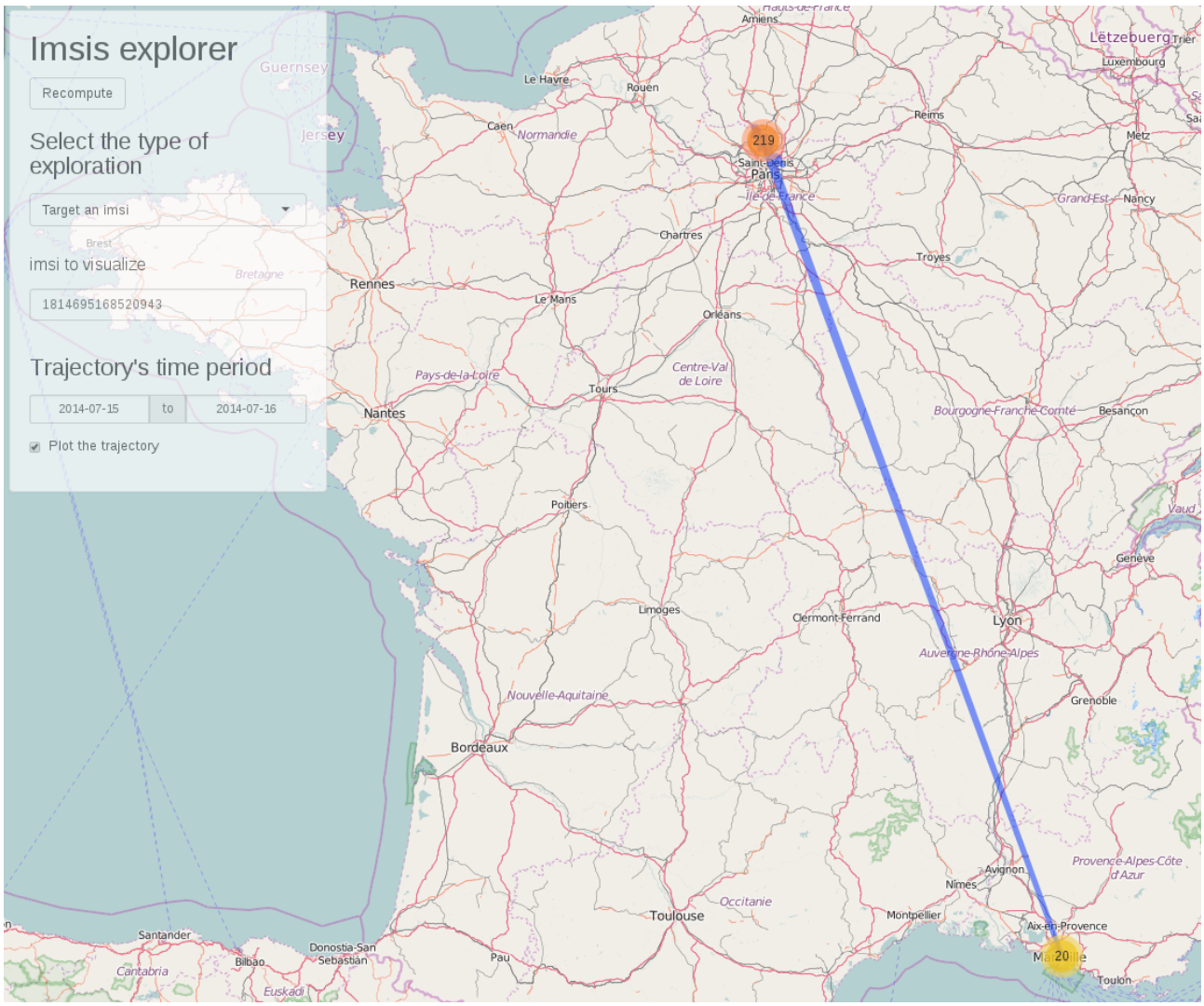
Peut-être la transformation spéciale pour le modèle gaussien explique une partie des différences entre les trois modèles. Mais il y a encore des spécialités sur chaque modèle qui ne sont pas visibles jusqu'à maintenant et qui restent à découvrir.

3.3 Quelques types d'anomalies

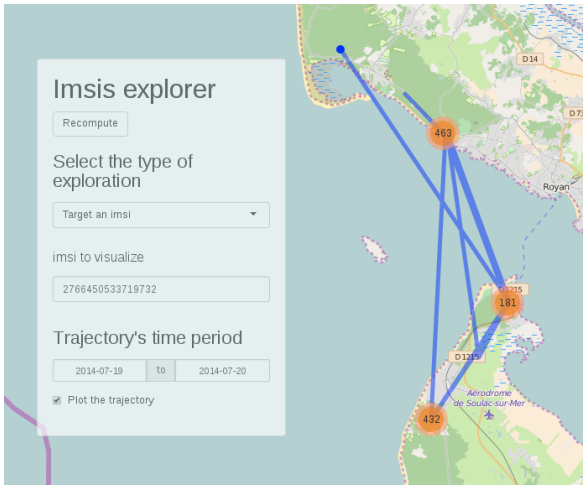
D'abord, les anomalies sorties ne sont pas forcément des imsi avec des comportements inhumains ou incompréhensibles. Ils peuvent être seulement un profil spécial mais tout à fait normal. Par exemple, dans la figure au dessous, c'est un imsi qui descend du nord au sud en voiture presque sans arrêts pendant une journée. Comme nous n'avons des données que sur deux jours pour cet imsi, la vitesse moyenne et aussi la moyenne du nombre d'événements chaque jour peuvent le rendre anormal par rapport aux autres imsi. Certainement, il devrait avoir d'autres types de profiles qui sont un peu moins visibles sur la carte.

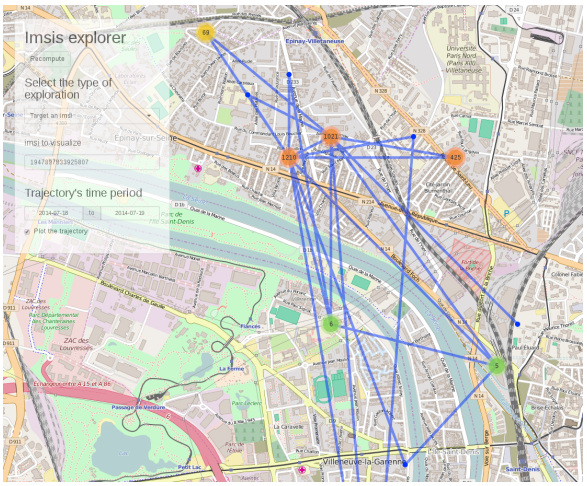


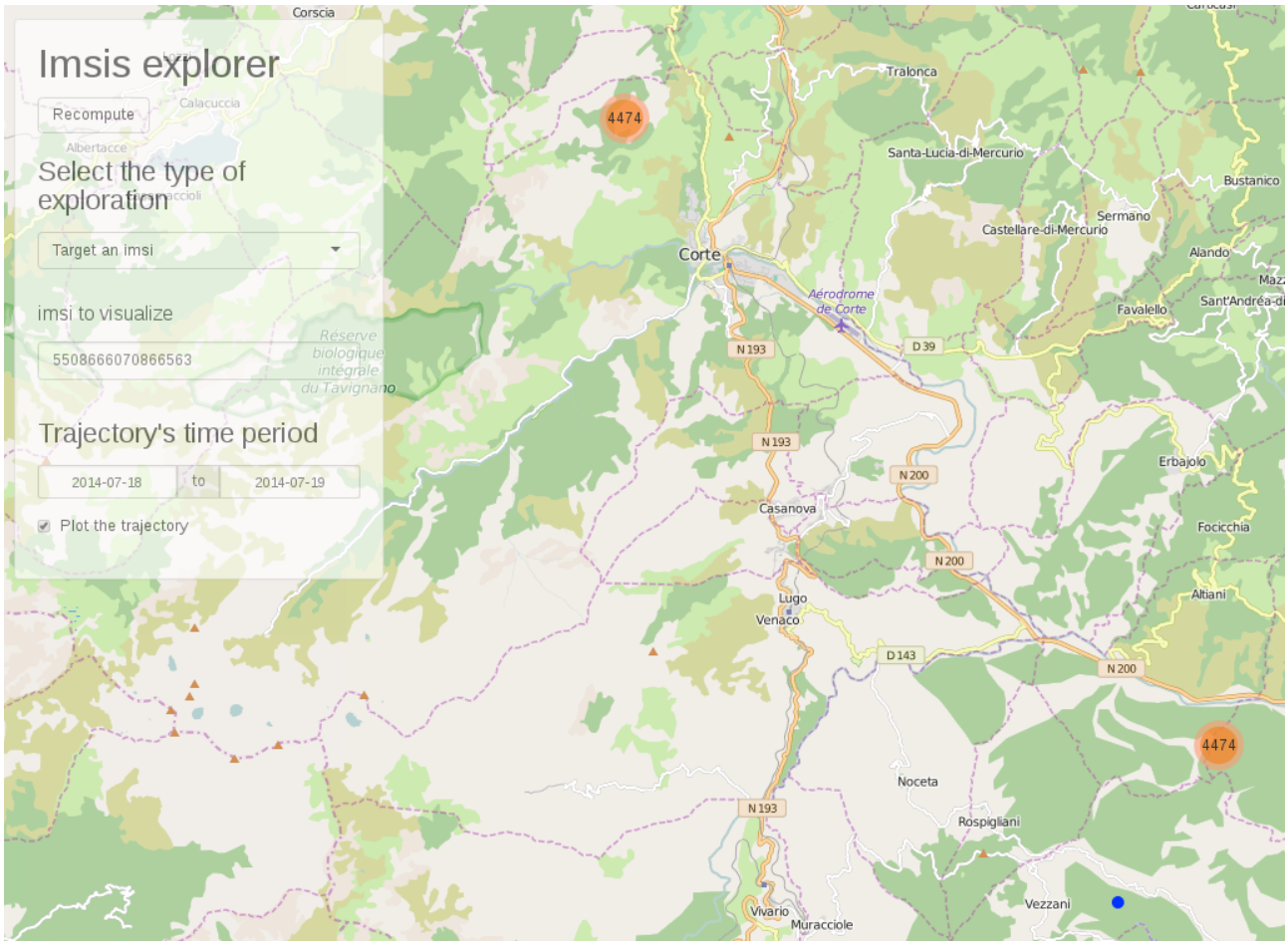
Deuxièmement, nous avons aussi des anomalies qui sont sorties à cause d'une cellule très mal positionnée. Dans ce cas, nous pouvons avoir un déplacement en quelques secondes avec une distance de plus de 100 kilomètres. Un exemple est donné par la figure suivante.



Sauf la vitesse maximale entre deux événements consécutifs, le nombre d'événements chaque jour peut exploser aussi. Dans ce cas là, c'est rare que les imsi allument vraiment des cellules partout pendant une journée. Mais plutôt quelques cellules sont allumées quelques centaines de fois. Ce genre de phénomène peut se passer dans une ville, dans un région de montagnes, où au bord de la mer comme les trois figures au dessous montrent. Les distances entre les cellules concernées peuvent être comparables ou beaucoup plus grandes par rapport aux rayons des cellules. Les comportements comme cela entraînent souvent aussi des très grandes vitesses moyennes.







Avec l'outil de visualisation que nous avons, les anomalies plus remarquables sont des imsi qui prennent les valeurs énormes sur la variable v_max et la variable n_events_mean. Comme les autres variables représentent mieux la mobilité d'une personne, les anomalies associées, par exemple à certains types de profiles, sont moins visible sur la carte. Avec un outil de visualisation plus avancé ou une analyse plus en détail, peut-être nous arriverons à déterminer des autres types d'anomalies.

3.4 Proposition

Pour que le résultat de la détection d'anomalies soit pertinent, les variables choisies et le modèle utilisé sont les deux parties les plus discriminantes. Ainsi, il vaut mieux faire encore

plus d'efforts pour comprendre les fonctionnements des trois modèles et choisir celui qui nous convient. En plus, plein de variables nouvelles pourraient être créées et examinées. Par exemple, nous pouvons construire des variables qui font la différence entre les jours dans la semaine et les jours de weekends. Dans le champs de type d'événement, nous n'avons utilisé que les données avec le type CELL_ENTER. Donc des variables qui prennent en compte des autres types d'événement pourraient être intéressantes aussi.