

Modélisation des trajectoires à partir de données radar (partie 1/2)

Cynthia Delauney
Laboratoire d'informatique de Paris 6 (LIP6)

September 4, 2017



Contents

1	Introduction	2
2	État de l'art	2
3	Rapide coup d'œil sur les données fournies par Safety Line	3
3.1	L'aéroport	3
3.1.1	Trajectoires	3
3.1.2	Pistes de décollages	3
3.2	Informations temporelles	4
3.3	Vitesses, Accélération, orientations	4
4	Notations	5
5	Trajectoires en retards - Modélisation du temps de roulage	6
5.1	En fonction des terminaux	6
5.2	Projection sur les heures	6
5.3	Projection sur les jours	6
5.4	En fonction de la météo	8
6	Modéliser l'ensemble du système	9
6.1	Représentation discrète de l'espace	9
6.1.1	Distribution des vitesses	9
6.1.2	Distribution jointe, direction et vitesse	10
6.2	Représentation continue	10
6.2.1	Réseau de neurones	10
6.2.2	Filtrage ?	12
6.2.3	Vraisemblance de la séquence : Analyse inter-trajectoire	12
7	Trajectoires non-avions	13
8	Annexe	14

1 Introduction

”Les données sur les trajectoires des avions sont depuis longtemps disponibles mais leurs exploitation s’est jusqu’à maintenant limitée à une utilisation en temps réel. Il n’y a pas eu jusqu’ici de tentative pour essayer de tirer parti de l’historique des déplacements sur les pistes et les voies de circulation, visant à apprendre à partir de situations réelles.”

Le but général est de **modéliser les trajectoires** d’avions sur l’aéroport en travaillant sur les deux axes : performance et sécurité. Le premier axe d’étude est celui de la performance, l’objectif est d’améliorer la circulation des avions sur l’aéroport. Les problèmes que l’on abordera seront ceux de la prédiction des temps de roulages, détection/prévention des situations de congestion, ... Le deuxième axe d’étude quant à lui permettra d’évaluer à quel point une trajectoire a été risquée. On peut résumer les deux axes en un problème de **détection de situations anormales** qu’on applique respectivement soit sur un ensemble de trace soit pour une trace plus un contexte. Cette étude apportera beaucoup car actuellement il est impossible de comprendre ce qu’il s’est passé suite à un événement sur l’aéroport, en effet les données récoltées (traces de tous les véhicules circulant sur la plate-forme : voitures, camions, avions, ...) commencent tout juste à être analysées.

Un rapide coup d’œil sur des données est fait, où l’on effectue des analyses statistiques sur les différentes *features* dont on dispose. Cela nous permettra de mieux cibler les données que l’on va utiliser pour chacun de nos futurs problèmes. Un grand nombre de ces *features* sont des données spatio-temporelles, on utilisera également certaines *features* tel que la configuration de l’aéroport. Le premier problème que l’on aborde est celui de vouloir prédire le temps que va mettre un avion à arriver sur une piste de décollage depuis son lieu de stationnement. Dans un premier temps on modélisera la **probabilité conditionnelle** du temps de roulage d’un avion conditionnellement à ses coordonnées initiales. On regroupera les trajectoires qui ont un comportement similaire en terme de distribution de temps de roulage pour ensuite interpréter les distributions obtenues (première détection d’anomalies : détection des trajectoires qui ont du retard). Après avoir considéré les trajectoires indépendamment, on va les considérer dans leur ensemble et chercher à caractériser l’espace dans l’objectif de détecter les zones de congestions (**représentation discrète** : découpage de l’aéroport en sous régions puis analyse des distributions de vitesses, des orientations, ...). On envisage alors la tâche d’annoter les régions de l’aéroport automatiquement (**k-means**, **Naive Bayes**). Afin de passer à une **représentation continue** de l’espace on aborde des modèles de réseaux neuronaux. Jusqu’à là le contexte pris en compte pour une trace était explicite, on va donc chercher à construire un contexte latent. Première projection des trajectoires dans un espace latent fait par comptage. Malgré la bonne interprétabilité de cette méthode son inconvénient est qu’il faut ajouter des colonnes pour ajouter des informations contextuelles, on s’intéresse donc plutôt à une projection latente avec ajout de contraintes, et donc un algorithme de type LVQ (learning vector quantization) est proposé.

2 État de l’art

Deux bibliographies :

- représentation légère d’une trajectoire
- mesure de similarité

3 Rapide coup d'œil sur les données fournies par Safety Line

On dispose de deux jeux de données, un jeu de données 'brut' et un 'nettoyé' (sachant qu'ici les données n'ont pas simplement été nettoyées, il y a de l'information ajoutée). Dans le jeu de données nettoyées, une trajectoire est représentée par plusieurs *features* ('smooth_speeds_cart', 'distance', 'smooth_coordinates', 'Vsmooth', ...). Dans un premier temps nous n'avons utilisé que les informations **spatio-temporelles** répertoriées dans la table 1. Toutes les études faites par la suite sont effectuées sur les données nettoyées, en ne gardant que les décollages.

<i>features</i>	domaine
mobile_type ⁽³⁾	{'aircraft', 'vehicle'}
duration ⁽³⁾	\mathcal{R} (unité : <i>seconde</i>)
trip_id ⁽³⁾	\mathcal{N}
datetime_start ⁽³⁾	"yyyy-mm-dd hh:mm:ss"
vsmooth ⁽¹⁾	\mathcal{R}^d , avec d le nombre de points que contient la séquence (unité : <i>m/sec</i>)
gsmooth ⁽¹⁾	\mathcal{R}^d , (unité : <i>m/sec</i> ²)
bearing ⁽¹⁾	Cap (modulo 360)
coordinates ⁽¹⁾	$\mathcal{R}^{d \times 2}$
nb_points ⁽³⁾	{20, ..., 31 871}
timesec ⁽¹⁾	\mathcal{R}^d (unité : sec)
distance ⁽¹⁾	\mathcal{R}^d (unité : m)
flight_action ⁽³⁾	{'None', 'landing', 'take-off', 'taxiing'}
runway ⁽³⁾	QFU de la piste pour les décollages et atterrissages

Table 1: Récapitulatif des *features* utilisées

3.1 L'aéroport

3.1.1 Trajectoires

Pour avoir une idée de ce qu'il se passe dans une journée, on a affiché les trajectoires de tous les engins (voitures, camions, avions, ...) circulant sur l'aéroport de Roissy Charles de Gaulle sur la figure 1. La trajectoire d'un avion qui va décoller représente une série de points variant généralement entre 700 et 3 000 points. **L'intervalle entre deux points varie et en terme de temps et en terme de distance**, mais globalement on a un intervalle de 0.5 secondes.

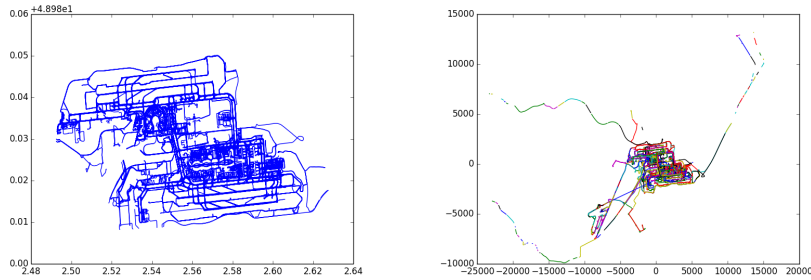


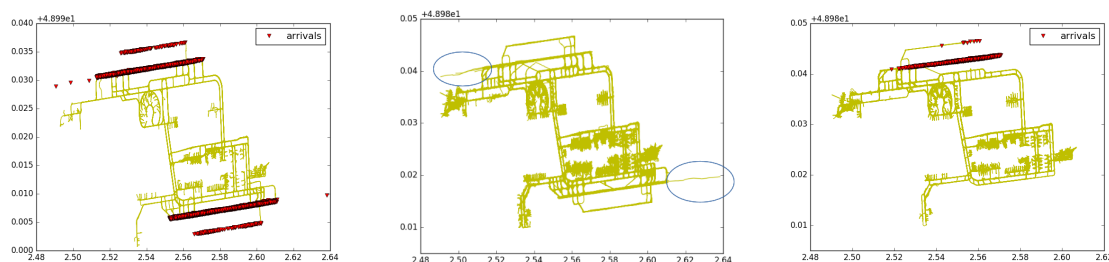
Figure 1: A gauche : Données nettoyées - Nombre de trajectoires au cours d'une journée : $\sim 3\,000$; A droite : Données pas nettoyées - Pas d'informations **mobileType**. Nombre de trajectoires : $\sim 10\,000$

Données sur 1 an : $\sim 130\,000$ décollages.

3.1.2 Pistes de décollages

Le point de décollage est la fin de notre séquence de coordonnées (il est difficile de savoir si la trajectoire s'arrête réellement quand l'avion quitte la piste...). On voudrait définir un point de fin de trajectoire commun à tous les décollages qui auront lieu sur la même piste. On cherche donc un point sur la piste, par exemple le point de décollage effectué avec la distance min, moyenne... Prendre l'entrée en piste n'est pas un bon repère, car l'avion peut se retrouver à attendre à ce moment là.

Remarques supplémentaires concernant l'aéroport :



(a) On affiche tous les points d'arrivés de toutes les trajectoires (b) Zones bleues indiquant des trajec- (c) Points d'arrivés des toutes les tra-
toires trop longues jectoires dont le $QFU = 09R$, on peut
voir qu'ici aussi des erreurs apparaissent

Figure 2: Comme on peut le voir sur les figures ci dessus, la fin de nos trajectoires ne correspondent pas au "décollage" (difficulté de définir exactement le point de détachement du sol : le capteur est plutôt "conservateur"). De plus pour certaines trajectoires elle s'arrête en dehors des pistes (zones entourées en bleu).

- Deux **configurations** (Est/Ouest) : les pistes qui sont à sens unique n'auront pas le même sens au cours de la journée, suivant la direction du vent, de plus les pistes changeront également de rôle (piste de décollage ou d'atterrissage), cela est du aux vérifications sur les pistes qui provoque leurs fermetures pour une durée déterminée

Sur un an (129 486 décollages) répartition selon la piste + configuration = QFU :

QFU	08R	09R	27L	26L	26R	27R	09L	08L
# décollage	105	15 450	38 031	128	45 241	152	42	30 337

Figure 3: Répartition de la variable **runway**

3.2 Informations temporelles

Une des *features* qui nous intéresse est **duration**, qui est le temps total de la trajectoire, on peut ainsi avoir une idée du temps de roulage d'un avion (le temps de roulage d'un avion est en général entre 10 et 15 minutes, voir figure 5).

Deux autres *feature* qui nous apportent une information de temps est **time** couplée avec **datetime_start**, **time** permet de voir le temps écoulé entre chaque points de la trajectoire, et **datetime_start** nous permet de savoir l'heure de départ de l'avion, on peut donc associer à chaque point une heure de la journée.

3.3 Vitesses, Accélérations, orientations

Les *features* **vsmooth**, **gsmooth** et **bearing** vont attribuer à chaque point la vitesse, l'accélération et la direction de l'avion θ . Les accélérations durant une trajectoire suivent les mêmes évolutions pour tous les décollages (une série de petites accélérations et de freinages, puis une grande accélération).

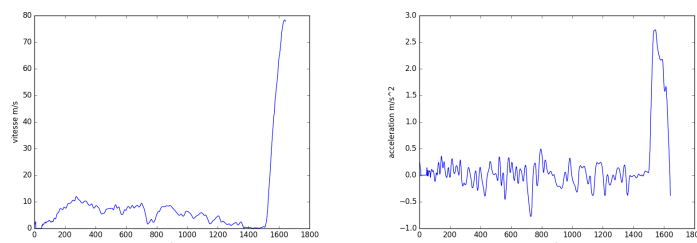


Figure 4: Évolution de la vitesse au cours du temps d'une trajectoire et évolution de l'accélération au cours d'une trajectoire

Remarque : au niveau des faibles vitesses, l'information de direction est très bruitée, comme on pourra le voir par la suite.

4 Notations

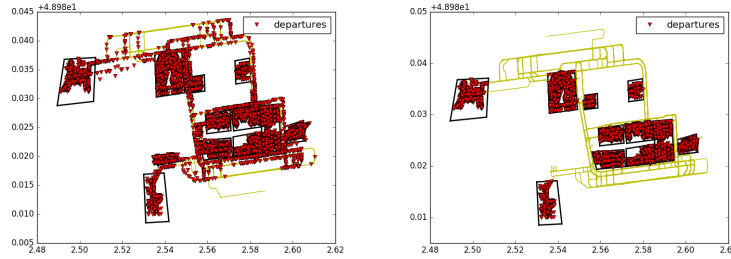
- on définit une trajectoire T_k par une séquence de coordonnées associé à un temps $T_k = \{\mathbf{c}, (t_1, \ell_1, \dots, t_{|T_k|}, \ell_{|T_k|})\}$, avec $(t \in \mathbb{R}, \ell \in \mathbb{R}^2)$ et c le contexte de l'avion
- pour chaque point (x_j, y_j) on connaît la direction de l'avion θ_j
- on note d_i la distribution des temps de roulage du cluster i
- \mathcal{T}_{delay} : trajectoires considérées en retard, récupérées depuis les distributions de chaque *cluster* (derniers centiles)
- \mathcal{T}_{prec} : trajectoires précurseuses d'anomalies
- \mathcal{R}_{delay} : régions où se croisent les trajectoires appartenant à l'ensemble \mathcal{T}_{delay}
- Z : dimension de l'espace latent ($nb_bin_grid \times nb_bin_grid \times dim_vitesse \times dim_direction$)
- avec la grille : une coordonnée l appartient donc à une région r
- S : dimension du vecteur associé à une trajectoire : vraisemblance $nb_bin_grid \times nb_bin_grid$

5 Trajectoires en retards - Modélisation du temps de roulage

Après avoir vu le comportement de certaines *features* caractérisant la trajectoire d'un avion au cours d'une journée, on pourrait comparer sur plusieurs jours si ces comportements sont toujours les mêmes, ou identifier des journées particulières. On a vu qu'au cours d'une journée on peut identifier des périodes de fort départ, c'est un premier indicateur naïf des moments de congestions dans une journée.

Les études qui suivent prennent toutes en compte la position de départ des avions. Pour réduire le nombre de points de départs on fait un clustering. Cette démarche permet de résoudre deux soucis :

- si l'on dispose de trop peu de données pour considérer la distribution du temps de roulage depuis chaque parking
- simplifier le problème en considérant les terminaux (= zones intéressantes), voir figure 19



5.1 En fonction des terminaux

On prends en compte uniquement les 11 terminaux représentés sur la figure 19, et on ne considère que les 4 principaux QFU. Pour chaque terminal on trace la courbe de distribution des temps de roulage, et on définit les trajectoires ayant eu du retard comme les trajectoires appartenant au **dernier centile** de la distribution (aucune **pondération** n'est faite) : on a au final 1 268 trajectoires, voir figure 6.

terminal_1	terminal_3	zone_attente_B1_B2	zone_fret_nord
16 202	5 889	748	3 475
zone_fret_sud	terminal_2B_2D	terminal_2F	terminal_2A_2C
1 262	15 117	32 869	9 904
terminal_2E	satellite_S3_S4	terminal_2G	
17 856	8 082	13 381	

Figure 5: Nombre de départs depuis chaque terminal

On constate que certains terminaux (comme le Terminal 2B 2D, le Terminal 2A 2C et le Terminal 2F) ont un comportement régulier quelque soit la piste de décollage, leur position semble idéal de ce point de vu. Certains terminaux n'ont pas beaucoup de départs et cela entraîne des courbes de distribution pas très lisses (comme pour les zones de fret et d'attente)

Remarques : Autre indicateur à explorer pour la congestion : 1) Météo (faible visibilité, dégivrage, ...), 2) Fermeture de voies (travaux, ...)

5.2 Projection sur les heures

On projette nos trajectoires de \mathcal{T}_{delay} sur les heures, on obtient ainsi une répartition des heures de départ dans la journée (voir figure 7(a)). Ces résultats sont intéressants notamment car on ne retombe pas sur les plages de hub (périodes de fort trafic) représentés en figure ??, en effet nos retards sont surtout concentrés vers 9h.

5.3 Projection sur les jours

On regarde la répartition des trajectoires en retard par jours, semaines, ou mois. Dans le cas où nos trajectoires en "retards" correspondent toutes à des perturbations qui ont eu lieu une même journée, il faudra au préalable travailler journée par journée pour ensuite les agréger en journées similaires et journées à perturbations (éventuellement ne pas les considérer par la suite).

Remarques :

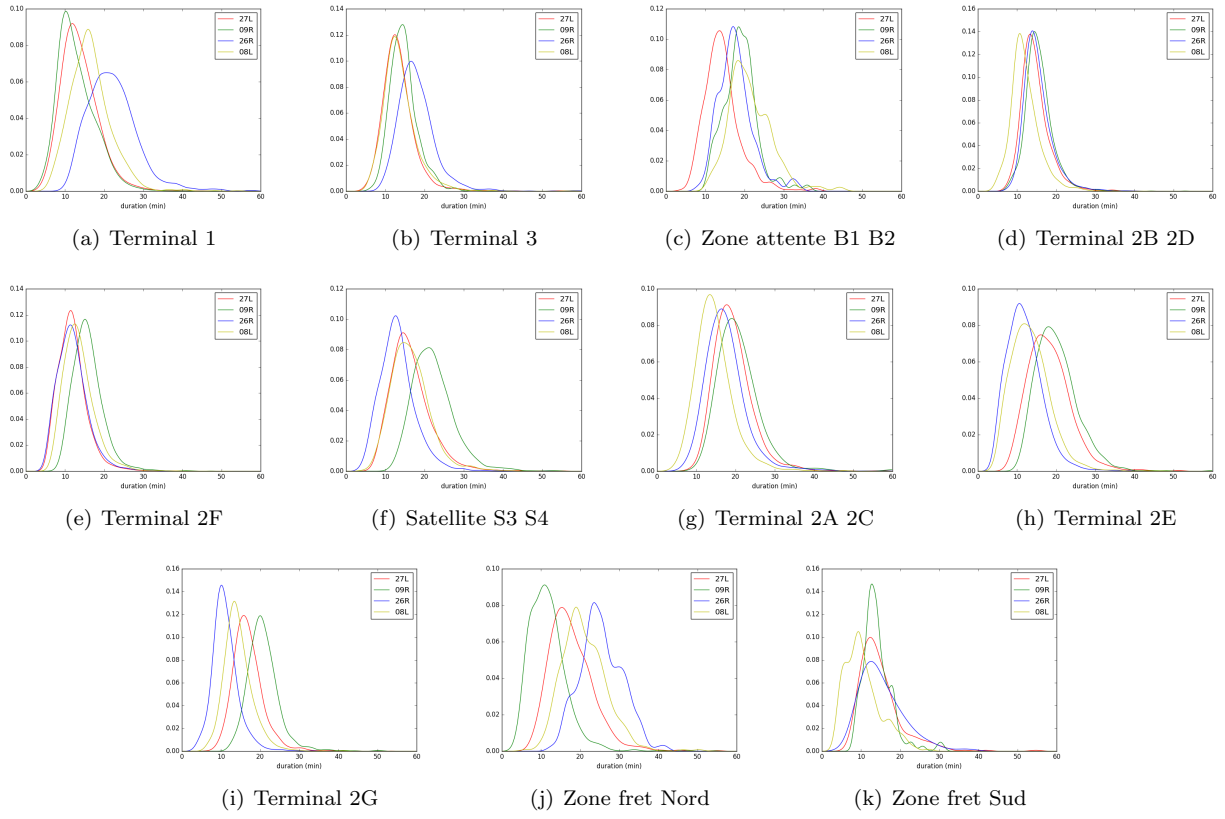


Figure 6: Distribution des temps de roulage en fonction du terminal conditionnellement à la configuration + piste de décollage

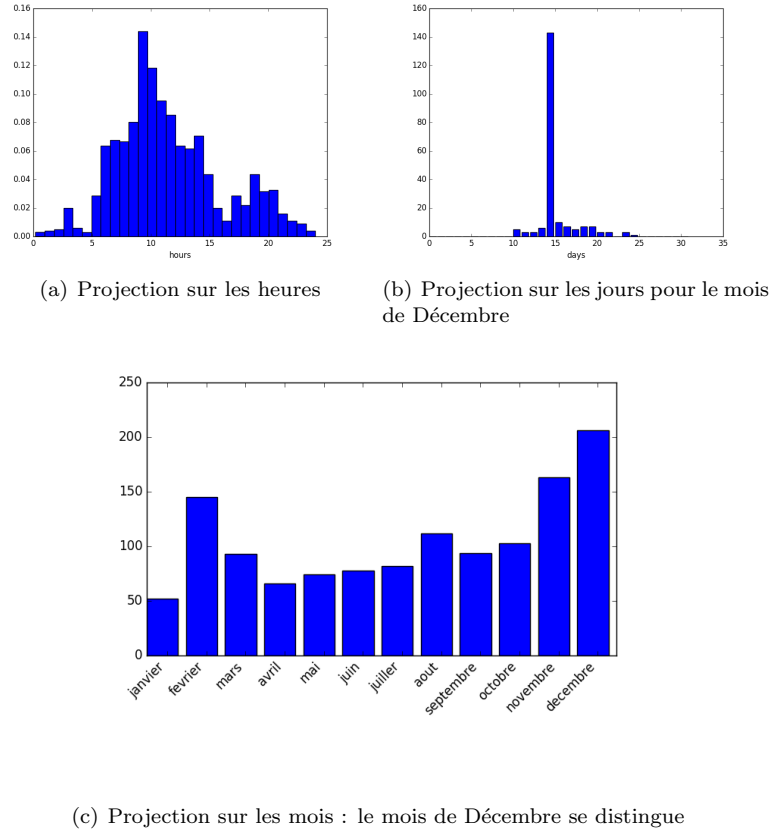


Figure 7: Périodes de fort retards, 1268 trajectoires constituent notre ensemble \mathcal{T}_{delay}

- Les trajectoires $\in \mathcal{T}_{delay}$ sont plus longues (en terme de nombre de point dans la séquence) que les autres. Cela pourrait simplement être du au fait que ces trajectoires sont plus longue (en terme de distance cette fois) que les autres, mais ici ce n'est pas le cas, les trajectoires sont à peu près équivalentes en terme de distance parcourue (vérifié avec la *feature distance*)
- $\sim 1\%$ des trajectoires sont retournées, si ce nombre semble trop restrictif, prendre le dernier décile et faire un *ranking* par exemple

5.4 En fonction de la météo



6 Modéliser l'ensemble du système

6.1 Représentation discrète de l'espace

6.1.1 Distribution des vitesses

On effectue un découpage dans lequel chaque région est de même dimension. La grille est découpée en $nb_bin_grid \times nb_bin_grid$, on affiche à titre d'exemple six régions différentes et la courbe de distribution des vitesses qui leur est associée sur la figure 8.

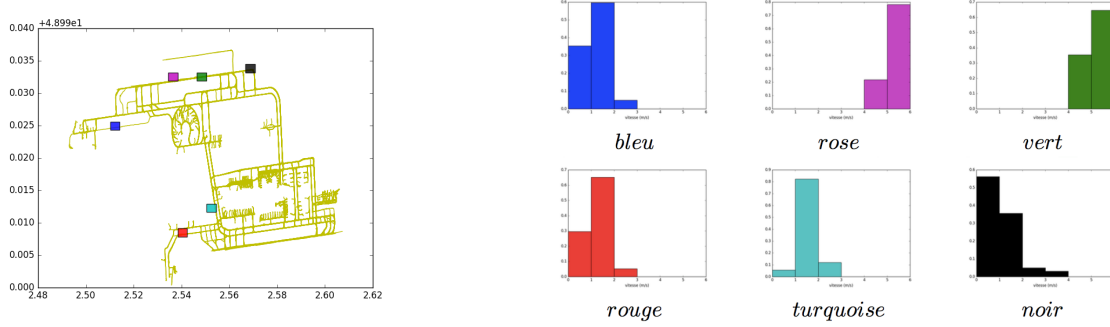


Figure 8: Représentation de quelques régions par leur distribution de vitesses

Un premier résultat de k -means sur ces histogrammes normalisés avec $k = 3$ nous semble déjà intéressant à interpréter. En effet les zones que l'on obtient sont les zones de parkings, voies et pistes (figure 9). Les zones d'entrées sur les pistes de décollages ont un comportement similaires que les zones de parking. Les vitesses vont en effet être les plus faibles dans ces deux régions, cependant une fois qu'un avion quitte sa position de stationnement il se s'arrêtera pas forcément, alors qu'en entrée de piste ce sera à chaque fois observé (avant décollage, quand l'avion est sur la piste il y a toujours un temps d'arrêt avant l'accélération finale qui mène au décollage). Pour palier ce problème, séparer les régions "entrée en piste" et "parking", on pourrait plutôt faire un k -means sur les distributions de nombre de stops dans la région (ou le temps d'arrêt dans la région), combiner les deux et utiliser un classifieur Naive Bayes. Sinon avec plus de données le problème sera également résolu, car en bout de piste les vitesses seront également très élevées et donc se distingueront des zones de parking.

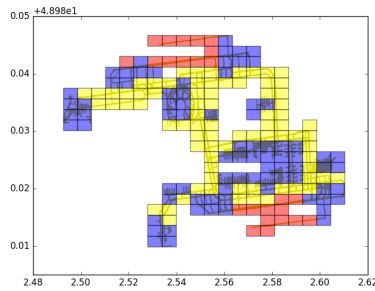


Figure 9: k -means sur les distributions de vitesses des régions, avec $k = 3$

Si on trace les trajectoires de \mathcal{T}_{delay} partant dans un intervalle de temps (congestion observée sur l'historgramme par exemple), on obtient des **nœuds clés**, ces régions constitueront notre ensemble \mathcal{R}_{delay} .

Au lieu de directement sélectionner un intervalle de temps, on pourrait regarder comment sont réparties dans la journée (avec forte concentration de retards) ces trajectoires : Analyse de la **résorption** d'une congestion sur une journée. Peut on reconnaître **différent type de congestion** ? Savoir si les avions se suivent ou se croisent. *Remarque : Prendre en compte les atterrissages ? Voir si il y a un lien au niveau des congestions avec les décollages, dans un premier temps faire séparément...*

6.1.2 Distribution jointe, direction et vitesse

On se propose maintenant d'analyser la distribution jointe de la vitesse et de la direction en chaque région.

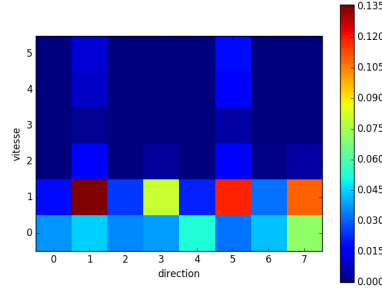


Figure 10: Sur l'ensemble des points

En tout il y a 48 classes, mais certaines classes ne sont pas représentées (36 classes représentées). Les directions sont discrétisées en huit (par angle de 45°) et sont représentées sur l'axe x , l'axe y correspond aux différentes classes de vitesse. Le pic qui est distingué est la direction entre 45 et 90° (sens anti-horaire) pour une vitesse entre 3 et 12 m/s . Sur les petites vitesses il n'y a rien de surprenant à constater qu'en effet on circule alors dans toutes les directions à peu près uniformément (de plus on a le soucis lié aux arrêts qui n'arrange pas ces changements de directions...).

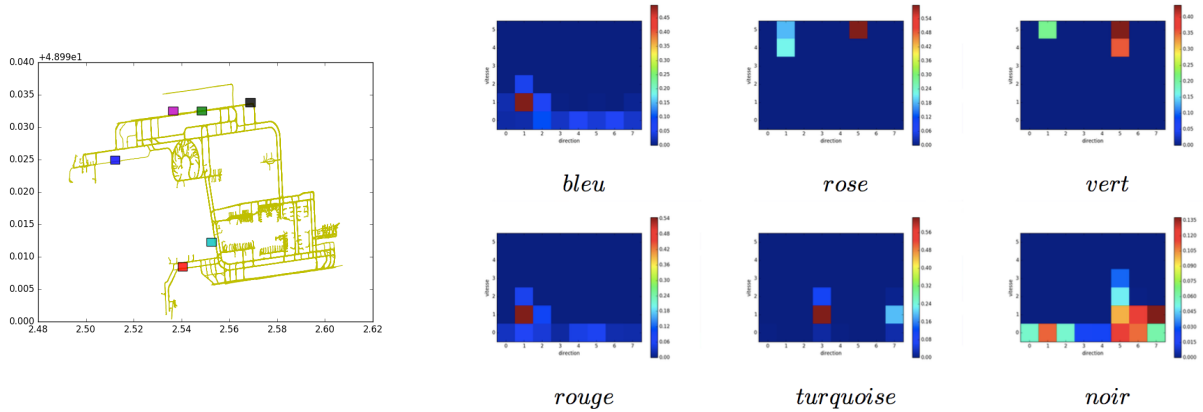


Figure 11: Distribution de la loi jointe sur quelques régions

De même que fait précédemment avec les distributions de vitesse en chaque région, on peut redéfinir l'espace en effectuant un k -means sur les distributions jointes caractérisant chaque région. On obtient alors une nouvelle visualisation de l'espace, ce qui nous permet de réduire le bruit dans les traces. Le point négatif de notre approche est la discrétisation faite par le découpage de la grille. Notons aussi que chaque point de la trajectoire est considéré de manière indépendante des autres points.

6.2 Représentation continue

6.2.1 Réseau de neurones

Le contexte peut être explicite ou latent. Dans un premier temps on ne considérera que les contextes explicites. Le réseau de neurones nous servira pour apprendre en chaque point la distribution de vitesses ou loi jointe vitesse/direction.

Propositions de modèles (contexte explicite):

1. Réseau de neurones avec en entrée $x_i, y_i, (t_i, \theta_i, acc_i, configuration)$, en sortie un histogramme H (vitesses, directions, ou loi jointe vitesse/direction) construit au fur et à mesure, pour l'apprentissage en sortie on donne un vecteur avec des valeurs nulles partout sauf à la classe vitesse correspondante. Ce réseau aura pour but d'offrir une **représentation de l'espace continue** en appliquant un k -means sur les vecteurs en sortie (on re discrétise ainsi les (x, y)).

Expériences

On appellera précision le pourcentage de bonne classification pour un échantillon tiré aléatoirement de la base de données. On compare les performances que l'on obtient par rapport aux performances obtenues avec d'autres modèles dans la figure 13.

Remarque : % de bonne classification calculée sur un sous échantillon, apprentissage et évaluation sont faites sur toutes la base de données (on cherche à faire du sur-apprentissage).

Modèle	Sortie	Hidden Units	<i>precision</i>	iterations
1	vitesse	[10]	0.47	50 000
2	vitesse	[30, 80]	0.73	1 000 000
2 bis (entrées en plus : t et θ)	vitesse	[30, 80]	0.79	1 000 000
3	loi jointe	[100]	0.19	50 000
4	loi jointe	[30, 80]	0.52	1 000 000

Figure 12: Résultats obtenus avec différentes structures de réseau de neurones

Sortie	Paramètres	% de bonne classification
vitesse	<i>LinearSVC</i>	0.52
vitesse	Régression logistique	0.50
loi jointe	Régression logistique	0.17
vitesse	Forêts aléatoires, profondeur = 5, 10 arbres	0.64

Figure 13: Résultats obtenus avec différents modèles

Les résultats montrent qu'en effet il est plus difficile de prédire les valeurs de la loi jointe. En complexifiant le modèle (en augmentant le nombre de couches cachées) on arrive à améliorer les performances. On les améliore aussi en ajoutant des informations de contexte.

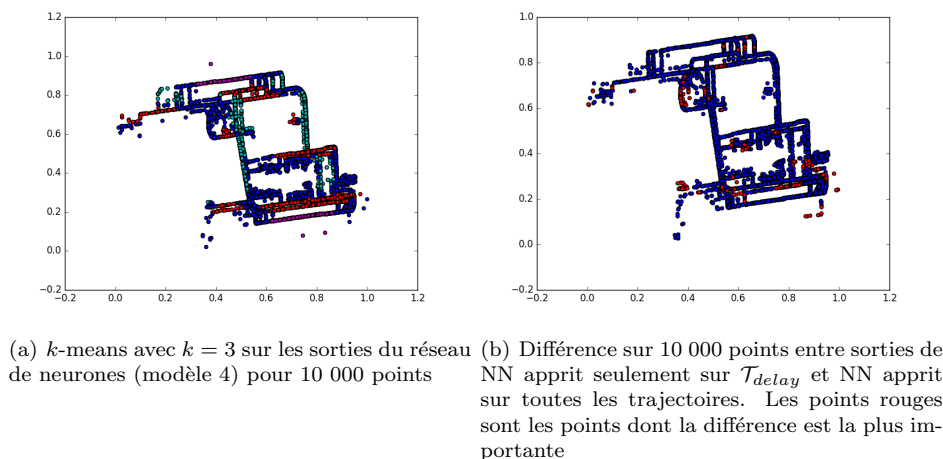


Figure 14: Utilisation des sorties du réseau de neurones

On obtient ainsi une représentation de l'espace uniquement à partir de nos traces. Précédemment nous avons considéré les régions de manière indépendantes les unes des autres, avec un grand nombre de données on peut considérer qu'avec le réseau de neurones les régions proches seront implicitement similaires. On voit qu'avec très peu d'information on arrive à définir des zones homogènes. On propose d'entraîner un réseau de neurones uniquement sur les trajectoires de \mathcal{T}_{delay} pour ensuite comparer les deux représentations de l'espace obtenues (c'est ce qu'on appelle faire du **Variational model**).

6.2.2 Filtrage ?

On peut filtrer certains points en regardant la valeur de l'**entropie** des distributions. Une entropie élevée signifie que le point est associé à une distribution pour laquelle les vitesses sont toutes à priori équiprobables, et ne nous aidera pas à caractériser notre espace. En localisant les points les moins informatifs selon la distribution des vitesses, on remarque que ce sont les zones sur les pistes où décollages et atterrissages se mélangent. Avec la distribution jointe les points les moins informatifs sont localisés au niveau de carrefour ou zone de parking.

6.2.3 Vraisemblance de la séquence : Analyse inter-trajectoire

On peut observer l'évolution de la vraisemblance au cours du temps, et détecter les trajectoires **précurseuses** d'anomalies (chute brutale de la vraisemblance), notons les \mathcal{T}_{prec} .

Amélioration sans régularisation spatio-temporelles : les points proches doivent avoir des sorties similaires... Comme on a beaucoup de points en entrée on suppose que cela se fait tout seul à condition de faire tourner le réseau de neurones assez longtemps pour qu'il sur-apprenne, sinon les points proches ne seront pas similaires (en terme de distribution en sortie du réseau).

Les courbes obtenues sont trop bruitées, en ajoutant d'autres informations de contexte (t_i et θ_i) on augmente les probabilités mais les courbes restent bruitées.

Axe absolu

Comme les trajectoires sont de longueur différente et que chaque trajectoire est échantillonnée différemment, la valeur de la vraisemblance d'une trajectoire par rapport à une autre n'a pas de sens. Pour comparer les trajectoires les unes avec les autres en regardant l'évolution de la log-vraisemblance on les affiche sur une figure avec un axe absolu pour le temps.

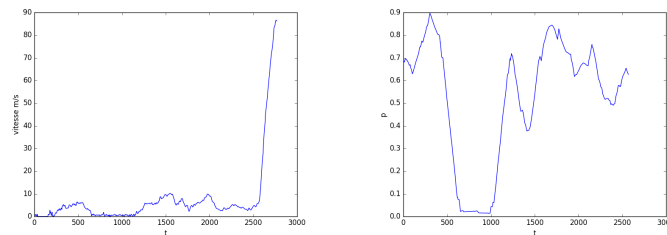


Figure 15: On détecte une période où la vitesse est anormalement basse, elle est associée à une faible probabilité

Le réseau de neurones nous aura finalement permis d'avoir une représentation de l'espace en continue. Le réseau de neurones nous aura également permis d'avoir une nouvelle représentation des trajectoires (évolution de la vraisemblance) mais pas exploitable pour le moment, besoin d'**améliorer les probabilité prédites** (faire tourner plus longtemps l'apprentissage du réseau de neurones ? filtrer les points ? Normalisation ?). Nous verrons qu'avec le découpage de la carte, on pourra utiliser une normalisation par région...

7 Trajectoires non-avions

On applique nos précédent modèles sur les traces véhicules. Les différents véhicules circulant sur l'aéroport sont répertorié sur la figure 16. Les différentes catégories de flottes possibles sont : Flyco, ATC equipment maintenance, Airport maintenance, Fire, Bird scarer, Snow Plough, Runway sweeper, Emergency, Police, Bus, Tug, Grass cutter, Fuel, Baggage, Catering, Aircraft maintenance, Unknown. Deux catégories sont distinguées en fonction de leur sensibilisation à la notion de **sécurité** :

- les véhicules jaunes : véhicules de service, véhicules de secours + véhicules de sûreté ('Flyco')
- les véhicules blancs : sous-traitants du gestionnaire d'aéroport (catering, bagages, véhicules de chantier)

'Emer-gency'	'Fire'	'Bird scarer'	'Flyco'	'ATC equipment maintenanc'	'Snow plough'	'Airport mainte-nance'	'Runway sweeper'	<i>Total</i>
8 695	2 737	4 074	6 030	12 609	1 233	3 364	1 633	40 375

Figure 16: Répartition des 44 326 véhicules selon leur tâches à effectuer sur l'aéroport (8 catégories observées)

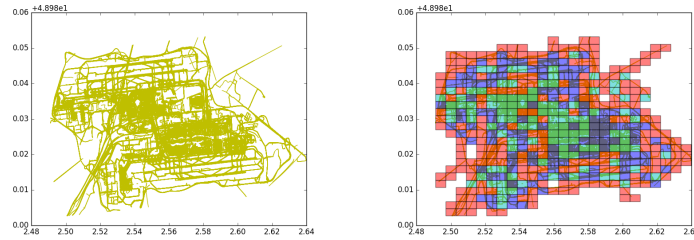


Figure 17: k -means avec $k = 3$ sur les histogrammes des vitesses pour la représentation discrète de l'espace

Ici $Z = nb_bin_grid \times nb_bin_grid \times 6$ (nous n'avons pas la variable d'orientation pour les véhicules...). Les trajectoires sont de longueur variant entre 20 et 78 924 points (en moyenne les trajectoires font 522 points).

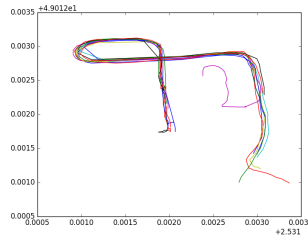


Figure 18: k -nn d'une trajectoire véhicule

8 Annexe

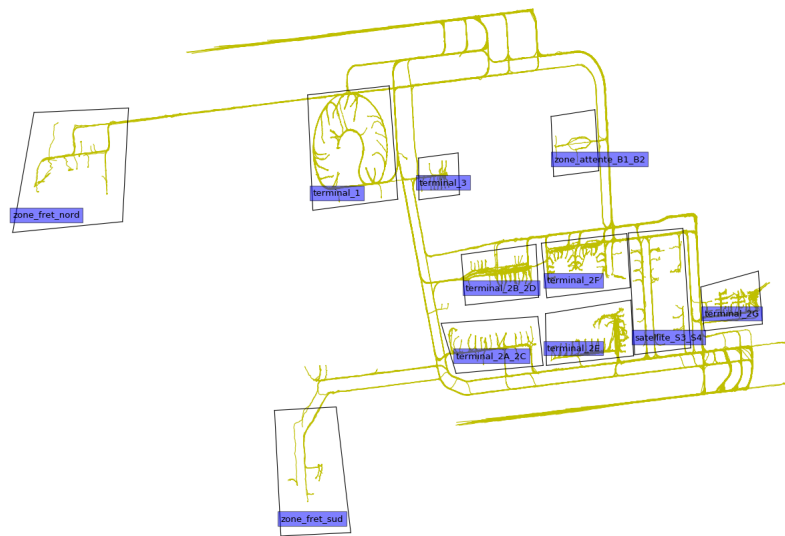


Figure 19: Les différents terminaux

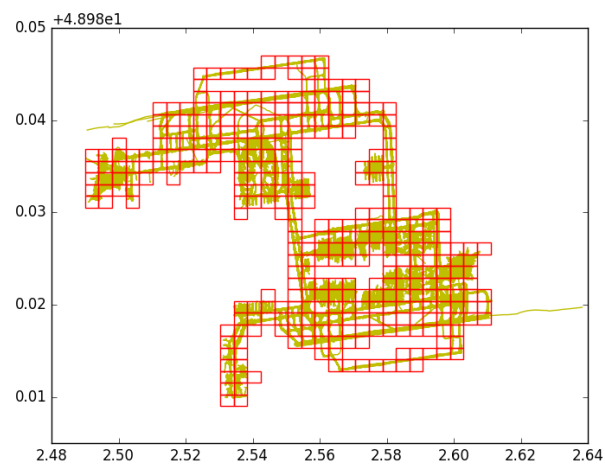


Figure 20: Découpage avec $nb_bin_grid = 30$, les points en dehors du grillage ne sont pas considérés