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

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

September 4, 2017

Contents

1 Introduction

Dans le précédent rapport une présentation des données a été faite, ainsi que des $features$ utilisées. On a vu comment construire l'ensemble $\mathcal{T}_{delay}$, ce qui nous servira ici encore.

Nos trajectoires vont être discrétiser de manière intelligente, en effet la projection des trajectoires brutes dans un espace intelligemment pensé nous permettra avec une métrique classique de regrouper des trajectoires similaires. Les différents espaces construit pour projeter nos données seront de même forme (cube), seules les valeurs prises différeront. Ce cube a donc de bonnes propriétés, notamment face aux bruits dans les données.

2 Modélisation d'une trajectoire dans un espace discret

2.1 Espace explicite discret

On projette nos données dans un espace explicite discret, nos trajectoire deviennent toutes de dimension Z. Dans cette espace on utilisera une métrique classique comme la métrique euclidienne. Cette représentation permet une **indexation intelligente** des trajectoires.

On projette nos trajectoires dans un espace $Z = nb\_bin\_grid \times nb\_bin\_grid \times 8 \times 6$. Une trajectoire est maintenant décrite par un vocabulaire où chaque mot correspond à un triplet région, vitesse, direction (r, v, d). On a donc maintenant $T_k = \{\mathbf{c}^\dagger, \mathbf{w}\}$ où l'information contextuel $\mathbf{c}^\dagger$ correspond à l'heure de départ de la trajectoire, ainsi que sa durée; et où $\mathbf{w} \in \mathbb{N}^Z$ est une représentation de taille fixe de tous les mots. Cette représentation des mots pourra être fréquentielle, binaire ou faite avec un tf-idf.

2.2 Les modèles

2.2.1 Projection Fréquentielle

On a maintenant $T_k = \{\mathbf{c}^\dagger, \mathbf{w}\}$ où $\mathbf{w} \in \mathbb{N}^Z$ est une représentation de taille fixe comptant le nombre d'occurrence de chaque mot. Dernière modification, on passe à une représentation par fréquence :

$$w_i^f = \frac{w_i}{\sum_j w_j} \in \mathbb{R}_+$$

$$T_k = \{\mathbf{c}^\dagger, \mathbf{w}^f \in \mathbb{R}_+^{\mathbb{Z}}\} \tag{1}$$

Remarques :

- On perd la notion d'ordonnancement mais ce n'est pas grave car on a l'information de direction dans notre représentation
- Semantic gap (réglé avec beaucoup de données)

Vraisemblance : Θ apprit sur toutes les trajectoires

$$\Theta = \begin{bmatrix} \vdots \\ \theta_i = p(w_i|\ell) \\ \vdots \end{bmatrix}, \Theta \in \mathbb{R}^{\mathbb{Z}} \tag{2}$$

$$p(w_i|\ell) = \frac{\sum_k w_i^{(k)}}{\sum_k \sum_{\{j|\ell \in w_j\}} w_j^{(k)}} \tag{3}$$

Le calcul de vraisemblance permet d'ajouter l'information de **normalité** ou non pour chaque point d'une trajectoire projetée dans l'espace discret. En utilisant le grillage on peut obtenir une mesure de vraisemblance dans chaque région ℓ et nous permet d'associer à chaque trajectoire T_k un vecteur de dimension $S = nb\_bin\_grid \times nb\_bin\_grid$, que l'on nommera $\mathcal{L}^k$.

$$\mathcal{L}_l^k = \frac{\sum_{\{i|\ell \in w_i^{(k)}\}} w_i^{(k)} \theta_i}{\sum_{\{i|\ell \in w_i^{(k)}\}} w_i^{(k)}} \tag{4}$$

Normalisation : Comme les cases n'ont pas les mêmes comportements, cad entropie différente (permet une visualisation de l'espace avec un *clustering* sur les distributions de chaque région, voir Rapport 1/2), on propose une normalisation par la valeur max de probabilité observée dans la case correspondante :

$$\bar{\mathcal{L}}_\ell^k = \mathcal{L}_l^k \times \frac{1}{\max\limits_{\{i|\ell \in w_i\}} \theta_i} \tag{5}$$

Chaque trajectoire est maintenant associée à un vecteur $\in \mathcal{R}^S$. La représentation d'une trajectoire prend maintenant la forme suivante :

$$T_k = \{\mathbf{c}^\dagger, \mathbf{w}^f \in \mathbb{R}_+^Z, \bar{\mathcal{L}} \in \mathbb{R}_+^S\} \tag{6}$$

On apprend huit modèles (pour chacune des valeurs de QFU, voir figure 18). **Finalement** on a un ensemble

$$\{\Theta^{08R}, ...\Theta^{08L}\}$$

qui va nous permettre d'associer à chaque trajectoire T_k sont vecteur $\mathcal{L}^k$.

2.2.2 Projection présentielle

On a maintenant $T_k = \{\mathbf{c}^\dagger, \mathbf{w}^b\}$ où $\mathbf{w} \in \{0,1\}^Z$ est une représentation de taille fixe décrivant la présence ou non de chaque mot.

$$T_k = \{\mathbf{c}^\dagger, \mathbf{w}^b \in \{0,1\}^{\mathbb{Z}}\} \tag{7}$$

On apprend cette fois Θ^b.

2.2.3 tf-idf

On a maintenant $T_k = \{\mathbf{c}^\dagger, \mathbf{w}^{tf-idf} \in \mathbb{R}_+^{\mathbb{Z}}\}$.

2.3 Use cases

Différent requêtage sont ici présentés, et feront parti des cas d'usage pour tous les autres modèles :

- depuis une trajectoire : on donne en requête une trajectoire complète, on cherche des trajectoires qui ont eu le même comportement depuis le départ jusqu'au décollage de l'avion
- une trajectoire + coordonnées géographique (direction et/ou vitesse) : comportement localement similaire dans une zone (ou des zones) à la trajectoire requête

On utilise la métrique euclidienne dans notre nouvel espace pour calculer les k plus proches voisins (k-nn).

2.3.1 Requête autour d'une trajectoire

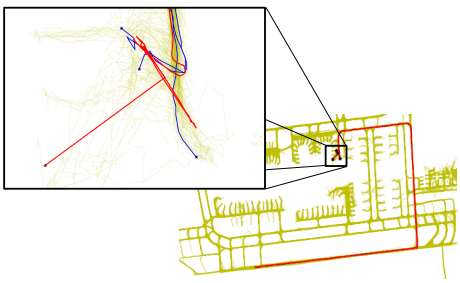
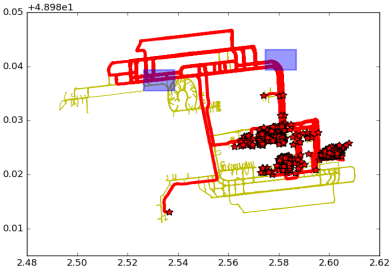


Figure 1: Requête en rouge, k plus proches voisins en bleues. On montre le bruit dans les données

2.3.2 Requête avec un ou plusieurs mots

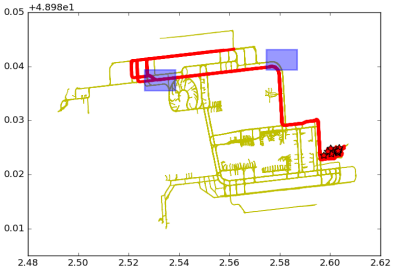
Les requêtes autour d'une trajectoire peuvent être vues de manière plus générale : comme des requêtes de mots. Les requêtes "**existentielles**" rendent un sous ensemble de trajectoires, on montre les requêtes sans pondération en Figure 2, et celles où on associe aux mots des valeurs de "présence" ou "non présence" en Figure A FAIRE. Un autre genre de requête est celle rendant un *ranking*, on associe ici des valeurs aux mots pour que les trajectoires rendues soient ordonnées selon leur similarité à la requête.

Requête "contenant" un des mots
$[(7, 8, 2, 7), (3, 7, 3, 5)]$



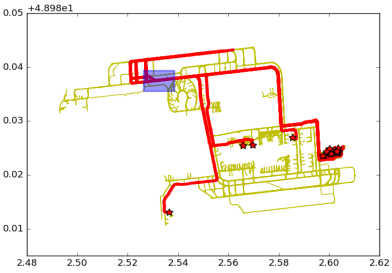
1 844 traj.

Requête "exactement" tous les mots
$[(7, 8, 2, 7), (3, 7, 3, 5)]$



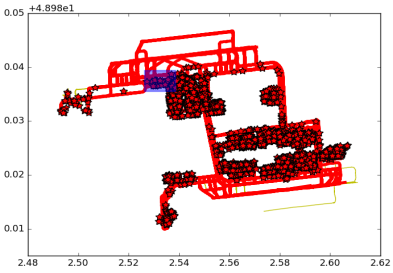
12 traj.

Requête "exactement" le mot (3, 7, 3,5)



21 traj.

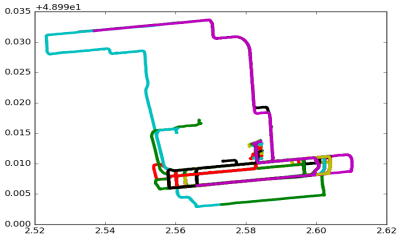
Requête "contenant" un des mots
$[(3, 7, -1, 5)]$



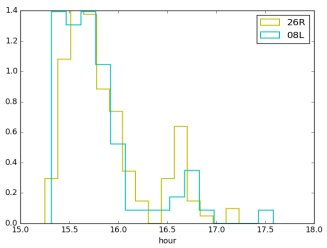
16 804 traj.

Figure 2: Les mots donnés en requête sont de la forme (i, j, v, d), pas de poids associés aux mots. On colore en bleu les régions contenant les mots

2.3.3 Vol régulier



trajectoires



heures de départs cond. aux deux principaux QFU

Figure 3: **callsign** = AFR1120, correspond à 234 vols

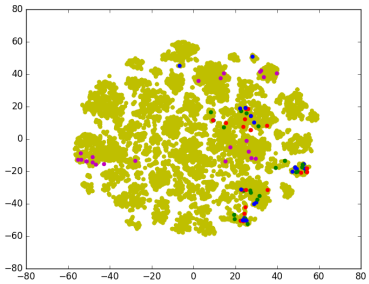


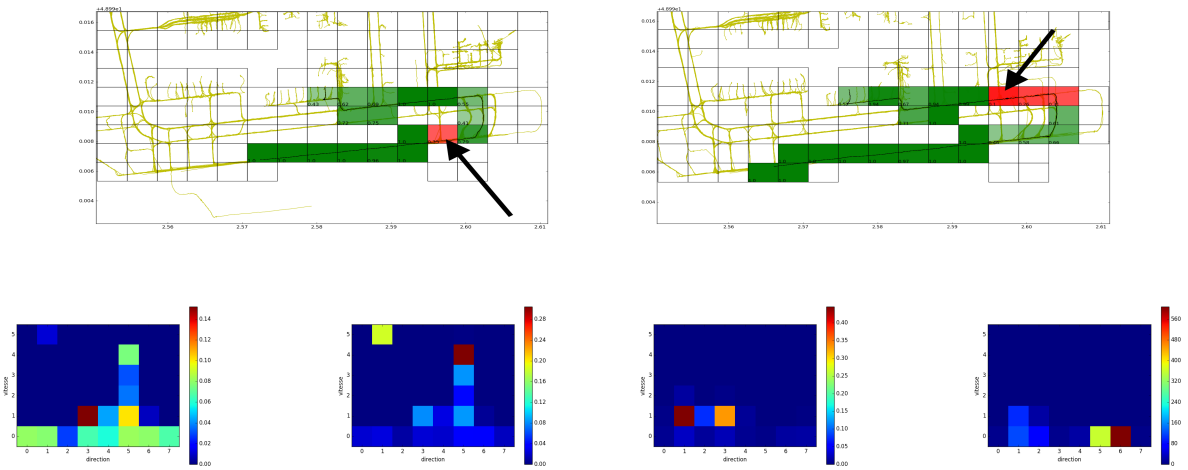
Figure 4: On affiche trois vols réguliers. En magenta vol régulier **AZ** les autres sont de svols réguliers **AFR**

2.3.4 Les trajectoires en excès de vitesse

Paramètre seuil à fixer : τ. Comme on construit un modèle par QFU on va extraire les excès de vitesse en fonction du QFU. On sélectionne les trajectoires pour lesquelles:

$$\exists\ j\ /\ \ell \in w_j, v_j > v^{\ell}_{ML},\ p(w_j|\ell) < \tau$$

2.3.5 Détection d'anomalie sur une zone



Valeurs selon Θ à gauche, à droite on calcule Θ en moyennant par trajectoire pour mettre en évidence que les points de faibles vitesses sont en sur-représentation

Valeurs selon Θ à gauche, distribution de la trajectoire dans la région à droite. On peut voir que les vitesses sont plus faible que la normal, et les directions sont bruitées

Figure 5: A gauche : région en entrée de piste montrée par la flèche à gauche, trajectoire normale. A droite : région avec la plus faible vraisemblance pointée par la flèche à droite, trajectoire en retard

2.4 Expériences

2.4.1 Comparer les modèles

On souhaite comparer les différents *ranking* rendus par nos modèles, afin de voir l'impact de la discrétisation de l'espace (grillage 10x10, 30x30, 50x50) et de la représentation choisies de nos données. Pour se faire on trace des courbes en fonction de la variable k (nombre de trajectoires rendues pour une requête), chaque point est calculé en moyennant les résultats de 100 requêtes aléatoires. Pour comparer le contenu des *ranking* rendus on calcul en chaque point le pourcentage de trajectoires en communs rendus par deux *ranking* (ratio) :

$$\frac{intersection(r^k_{m1}, r^k_{m2})}{k}, \quad r^k_m \text{ ranking rendu par le modèle } m$$

La Figure 6 compare donc les résultats rendus par les modèles binaire et tf-idf par rapport au modèle fréquentiel. Les modèles tf-idf et fréquentiel rendent des résultats très similaires. Le modèle binaire quant à lui rend des résultats différents dés le début, mais qui s'en approche avec un grillage de plus en plus fin.

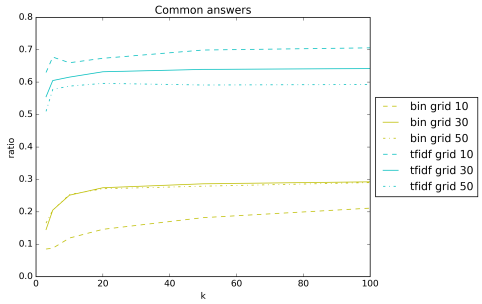


Figure 6: Comparaison des résultats **communs** retournés par rapport à la représentation fréquentielle pour différente discrétisation

On peut maintenant chercher à savoir à quel point les trajectoires rendues par les différents modèles colleront à la similarité géographique de la trajectoire. On utilise une mesure de similarité entre deux trajectoires reposant sur le passage ou non dans les mêmes régions. On observe le même comportement pour tous les modèles (Figure 7) : les réponses sont progressivement de plus en plus dispersées, ce phénomène est plutôt lent.

$$sim(traj_1, traj_2) = \frac{intersection(regions(traj_1), regions(traj_2))}{union(regions(traj_1), regions(traj_2))}$$

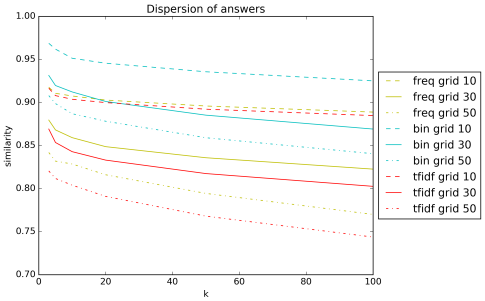


Figure 7: Comparaison de la **similarité** des résultats à la requête

Remarque : il est normale que nos valeurs soit plus petites avec un grillage plus fin, on devient encore plus sensible aux faibles différences

Nos modèles vont donc favoriser des réponses se superposant à la requête, mais lorsque la trajectoire requête correspond à une anomalie ce ne sera pas autant le cas comme montré en Figure 9. Pour les retards on observe une plus grande dispersion dans les réponses (la région où le retard a lieu prédomine), et ce sont les modèles plus sensible à un événement long dans une région qui captureront mieux les retards (freq et tf-idf).

Pour les trajectoires en excès de vitesse on remarque que de manière général ses voisins seront un peu plus éloignés, mais le score est assez faible et semble plus sensible au grillage choisi : en effet on remarque que ce n'est pas un grillage 50x50 qui rend un meilleur score mais le grillage 30x30. Le modèle binaire rend des trajectoires collant mieux à la requête et lui permet d'obtenir un meilleur ratio.

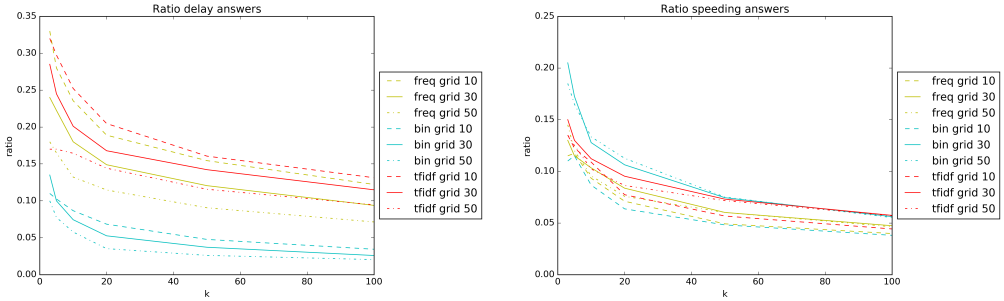


Figure 8: Comparaison sur la capacité de chaque modèle à rendre des trajectoires anormales, en gauche les retards et à droite les excès de vitesse.

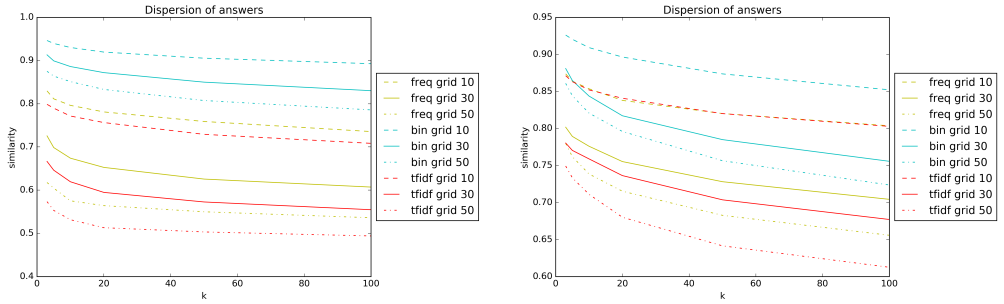


Figure 9: Dispersion des réponses à une trajectoires requête anormales. Retard à gauche, excès de vitesse à droite

2.4.2 Aspect Sémantique

Sémantique

On définit le voisinage d'un mot par :

- ses 8 voisins dans l'espace (au max) + 2 voisins de vitesse (au max) + 2 voisins de direction (exactement)

Pseudo relevant feedback : Une alternative est de redonner en requête la requête et ses réponses.

2.4.3 T-SNE

Une première manière de comparer les différents modèles est de voir comment les trajectoires se répartissent dans chacun des espaces, et notamment les trajectoires particulières telles que les retards et les excès de vitesse. La réduction de dimension faite avec l'algorithme t-sne nous permet de visualiser en deux dimensions les trajectoires. Il y a quatre pistes de décollage, donc huit valeurs possibles : QFU $\in$ {08R, 09R, 27L, 26L, 26R, 27R, 09L, 08L}, cependant on en affiche que les 4 principaux.

Le cube utilisé est réduit, pour des soucis de calculabilité, on découpe la carte avec un grillage 10×10. Les trajectoires que l'on considère en retard ne sont pas regroupées dans un seul et même *cluster*, mais on distingue des petits regroupements, Figures 10. Le modèle binaire est un peu moins bon que les deux autres. Les excès de vitesse quant à eux ne ressortent pas avec le t-sne, comme un excès de vitesse peut être très localisé il est difficile de valoriser son impact, voir courbes Figure 8.

Les retards vont se retrouver au milieu des gros *clusters*, car une même région ayant souvent du retard correspond souvent aux entrées de piste et ces trajectoires proviennent de différentes zones de parking. Ici ce n'est pas tout le temps de cas, mais on verra qu'en sélectionnant de manière plus fine les trajectoires en retard on arrive à mieux visualiser cette remarque.

2.5 Autres conditionnements possibles

Précédemment on a conditionné nos modèles apprit selon la variable **runway** (information QFU), mais on peut également conditionner selon :

- meteo normale ou dégradée
  - LVP (Low Visibility Procedures) : décollages par faible visibilité

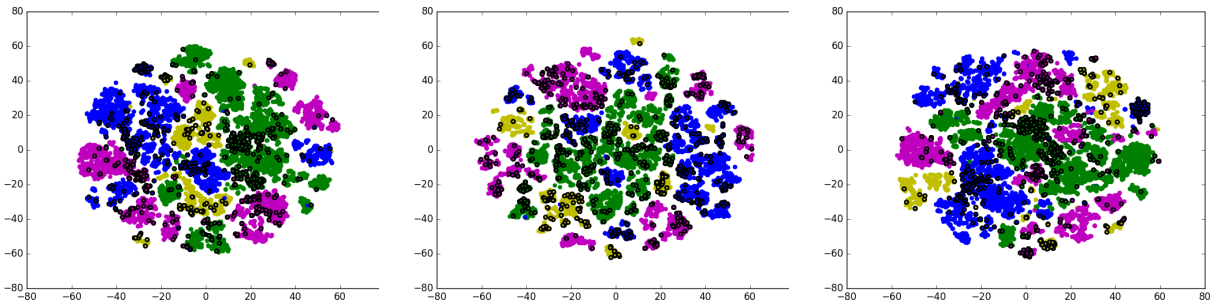


Figure 10: t-sne, modèle fréquentielle, binaire et tf-idf, les trajectoires en retard sont entourées (1 270). On colore les trajectoire selon leur QFU ; en rose : 08L, en jaune 09R, en vert 26R et en bleu 27L

  - les avions qui passent par des baies de dégivrage doivent y rester entre 10 et 20 minutes

- type d'avion : ils n'utilisent pas les mêmes pistes

	lower_heavy	upper_medium	upper_heavy	lower_medium	super_heavy	light	None
Total	6 618	**72 934**	**23 435**	**23 485**	2 599	313	132
26R	1 245	26 454	7 721	9 021	775	60	0
08L	1 170	18 057	5 656	4 813	578	70	3
27L	2 761	20 464	7 377	6 385	902	117	20
09R	1 443	7 860	2 638	3 082	346	59	10

Figure 11: Répartition des différent type d'avion selon le QFU

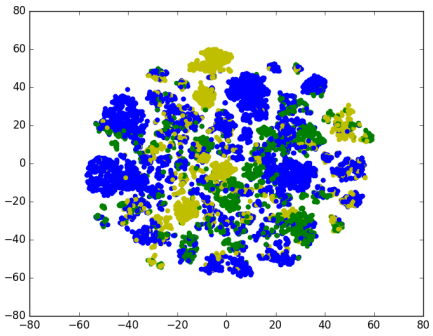


Figure 12: t-sne, on colore les 3 principaux types d'avions. En bleu : upper_medium, vert : upper_heavy et en jaune : lower_medium

	Snow	Fog	Rain	Other	Total
	2 198	13 697	**68 374**	55 530	129 516
$\in T_{delay}$	60	**433**	610	167	1 270

Figure 13: Répartition des avions selon la météo (*wunderground* : *dailysummary*). Certaines trajectoires peuvent avoir plusieurs variables à vrai, Other correspond à des trajectoire qui ont aucunes de ces valeurs à vrai

Comme première analyse on récupère seulement l'information *dailysummary* : pour ce qui est de la neige cela semble raisonnable de généraliser à la journée, mais pour le brouillard c'est incorrecte, en effet une condition critique du au brouillard sera de courte durée. En coloriant sur le t-sne les trajectoires ayant eu lieu pendant des journées définies de brouillard on ne distingue rien, alors que les trajectoires ayant eu lieu pendant des journées de neige forment des sous-*clusters*. On remarque que les trajectoires **Snow** et **Fog** correspondent à 1/3 des retards.

Les passages aux **baies de dégivrage** (dans les trajectoires $\mathcal{T}_{delay}$) :

- attente supérieur à 10 min dans les zones définies : 197 traj
- attente entre 10 et 20 min dans les zones définies : 175 traj
- au dessus de 20 min : la plupart entre 20 et 25 min, 3 au dessus de 40 min

En tout 343 traj passent plus de 10 min dans les baies, et 313 traj y passant entre 10 et 20 min.

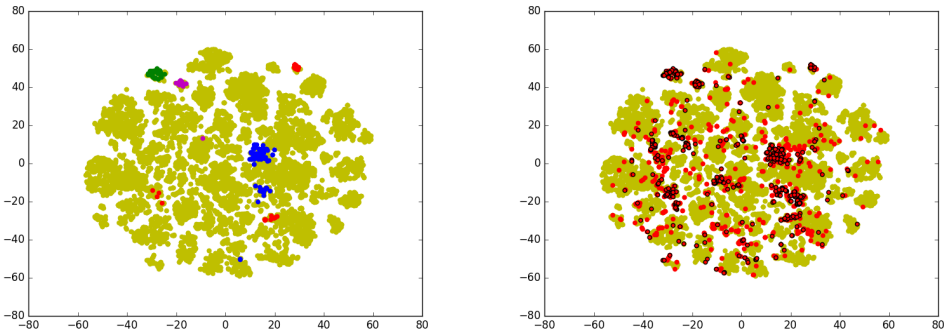


Figure 14: t-sne, codage fréquentiel. **A gauche :** on colore les trajectoires passant par des baies de dégivrage, y restant entre 10 et 20 min. En bleu : zone_SE, en vert : zone_NE, en rouge : zone_SW et en rose : zone_NW. **A droite :** retards en rouge vs snow and fog qui sont entourées en noir

Ces longs moments d'attente ont un poids important avec un codage fréquentiel de l'information, et permettent de regrouper les trajectoires prenant du retard dans la même région, contrairement au codage binaire par exemple. On redéfinit nos trajectoires en retards en ne se basant plus sur toutes les trajectoires mais seulement sur les trajectoires n'appartenant pas à des jours de neige ou de brouillard, et ne passant pas par les zones de baies de dégivrage plus de 10 min. On notera $\mathcal{T}^*_{delay}$ cet ensemble de trajectoires. L'affichage du t-sne restera semblable, car des retards auront lieu dans des régions où se trouve également les zones de dégivrage (surtout avec un grillage 10x10) : c'est a apprenant des nouveaux paramètres qu'on constatera que les trajectoire passant par les baies de dégivrage ne seront plus considérées anormales (trajectoire à droite en Figure 5 qui est considérée anormales dans 3 régions successives : avec de nouveaux paramètres ce n'est plus le cas comme illustrée en Figure 16)

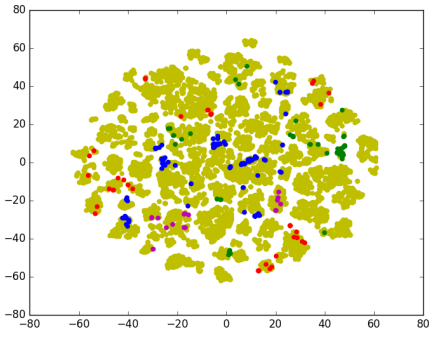


Figure 15: t-sne, codage binaire. On colore les trajectoires passant par des baies de dégivrage, y restant plus de 10 min

Figure 16: Une trajectoire passant par une zone de baie de dégivrage

3 Contextualisation

Dans la partie précédente on a vu comment détecter un comportement anormal, maintenant on aimerait détecter des situations anormales

Les seuls informations de contexte que l'on avait avant étaient l'heure de départ et le temps de roulage de chaque avion. On ajoute a notre vecteur représentant une trajectoire un autre vecteur représentant le contexte de la trajectoire ($dim\_context = nb\_bin\_grid\_v \times nb\_bin\_grid\_v \times 6$, où $nb\_bin\_grid\_v$ est le découpage de la carte pour les véhicules, pour le moment les véhicules n'ont pas d'infirmation de direction).

Dans un premier temps on se focalise sur les **incursions sur piste** : *"Toute situation sur un aérodrome entraînant la présence inopportune d'un aéronef, d'un véhicule ou d'une personne dans l'aire protégée d'une surface destinée à l'atterrissage ou au décollage d'aéronef."* (def. OACI), et donc on ne s'intéressera qu'aux dernières traces véhicules (**1 minute avant le décollage**) dans la région de la piste d'où décolle l'avion. Agrégation de ces nouveaux vecteurs latent ? Requête uniquement sur le contexte dans ce cas ? Comme le vecteur contexte est très *sparse* on propose de faire une passe de **filtre** pour accentuer les directions et vitesses.

Figure 17: Contexte d'avion circulant le 2015-11-14, **datetime_start** = 07:36:38. Trajectoire de l'avion en bleu et en rouge celles des véhicules

Comme ce qui nous intéresse ce sont les véhicules à proximité des avions sur la piste de décollage et en terme d'espace, et en terme de temps, une autre manière de voir les choses et d'ajouter l'aspect **temporelle** du contexte de la manière suivante : en chaque point de la trajectoire est associé le contexte (On additionne plus les vecteurs contextes mais on les multiplie, on explose la dimension de notre espace latent ! $Z = nb\_bin\_grid \times nb\_bin\_grid \times 8 \times 6 \times dim\_context$). Possibilité de pondération ?

L'avantage de cette méthode est l'interprétabilité des colonnes. Cependant on doit ajouter des colonnes pour ajouter des informations contextuelles. L'espace ainsi construit devient vraiment grand, c'est une limite, en effet on aimerait affiner le plus possible le grillage de la carte et ajouter des informations contextuelles.

4 Modélisation d'une trajectoire dans un espace latent continu

4.1 Learning vector quantization

On veut donc généraliser l'approche que l'on effectue précédemment et notamment ne plus avoir à effectuer une discrimination de l'espace. Ce travail nous permettra de prendre en compte diverses connaissances métiers par l'ajout de contraintes, qui sont d'ailleurs des contraintes hétérogènes (avion, véhicules, météo, horaire, type de journée, piste...) et rend encore plus intéressant ce travail.

Modélisation latente des trajectoires + contexte

1. **critères** (multiples) rapprocher/éloigner
    - en fonction du contexte
2. **projection** d'une population + détection des retards
3. **analyse** des retards dans l'espace latent

Algorithme LVQ :

1. [Init.] Projections aléatoire des trajectoires sur $\mathbb{R}^Z$
2. Tant que la convergence n'est pas atteinte
    (a) Tirer aléatoirement 2 trajectoires i et j
    (b) Les rapprocher/éloigner en fonction de différentes contraintes

Quelques idée de critères pour rapprocher les trajectoires : si passage dans la même région avec un même comportement, si temps de roulage équivalent...

5 Annexe

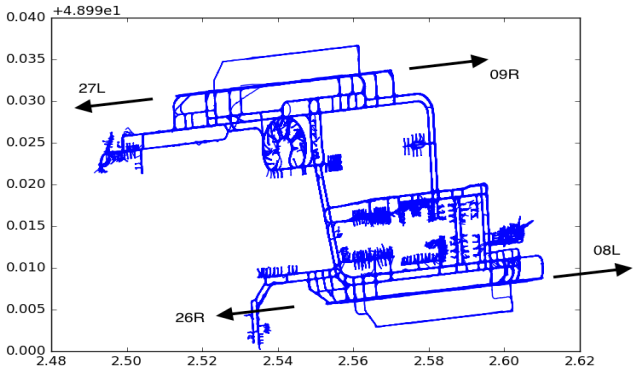


Figure 18: Voies empruntées par les avions