Reconnaissance des modes de transport par apprentissage profond à partir de signaux GPS

ANDREA VASSILEV

CEA, LETI, MINATEC Campus, F-38054 Grenoble, France
andrea.vassilev@cea.fr

Résumé –

La diffusion de plus en plus large de dispositifs portables (smartphones, montres connectées, …), intégrant une multitude de capteurs (accéléromètre, magnétomètre, gyromètre, GPS, …) conduit à un nouvel intérêt pour les applications de capture de contexte, comme par exemple la reconnaissance automatique du mode de transport utilisé par une personne. Cette problématique a déjà fait l'objet de nombreux travaux qui reposent sur des approches de classification supervisée 'traditionnelles', i.e. en construisant manuellement les descripteurs ('features'). Suite au développement récent de l'apprentissage profond, il semblait intéressant d'évaluer quel peut être l'apport de cette méthode à notre problématique. Nous avons donc développé un réseau convolutionnel profond sur la base de données publique Geolife constituée de signaux GPS. La performance obtenue, de l'ordre de 79% est supérieure de 3% à celle obtenue par une approche traditionnelle. Nos principales contributions ont été de mettre en évidence que les signaux les plus pertinents sont la vitesse et l'accélération, que la durée des signaux en entrée du réseau doit être suffisante, de l'ordre de 17 min et que certains signaux présentent une autocorrélation importante qui peut conduire à une surestimation des performances du modèle, de l'ordre de 5%, si la partition des données n'est pas réalisée avec soin.

Abstract –

The diffusion of an increasing number of portable devices (smartphones, connected watches, ...), embedding a multitude of sensors (accelerometer, magnetometer, gyrometer, GPS, ...) leads to a new interest for context capture applications, such as the automatic recognition of the mode of transportation used by a person. This problem has already been the subject of many researches based on 'traditional' classification approaches, i.e. manually constructing the features. Following the recent development of deep learning, we developed a deep network on the Geolife public database. The performance obtained, about 79%, is 3% higher than that achieved by a traditional approach. Our main contributions have been to demonstrate that: the most interesting signals are speed and acceleration; the duration of the input signals of the network should be sufficient, about 17 minutes; and due to the strong autocorrelation of input signals, the performance of the model can be overestimated by up to 5%, if the data selection is not done carefully.

1 Introduction

1.1 Contexte

La diffusion de plus en plus large de dispositifs portables (smartphones, montres connectées, …), intégrant une multitude de capteurs (accéléromètre, magnétomètre, gyromètre, GPS, …) conduit à un nouvel intérêt pour les applications de capture de contexte, p. ex. la reconnaissance du mode de transport utilisé par une personne. Les applications sont nombreuses : estimation de l'empreinte carbone, enquêtes déplacement, analyse de la façon de conduire, …

Cette problématique a déjà fait l'objet de nombreux travaux [1] [2] [3] qui reposent sur des approches de classification supervisée 'traditionnelles' ; celles-ci présentent les inconvénients de nécessiter une expertise pour la conception des descripteurs, d'être gourmand en temps de conception et d'implémentation, et d'être potentiellement sous optimale puisque les données brutes sont condensées en un nombre restreint de variables.

L'apprentissage profond supervisé est une approche opposée, dans laquelle, les données brutes sont présentées en entrée du modèle et les descripteurs pertinents sont en principe appris par le modèle. Les réseaux convolutionnels profonds sont un exemple qui a récemment obtenu des résultats impressionnants en reconnaissance d'images. Les objectifs de cet article sont d'évaluer l'apport de l'apprentissage profond sur la base de données publique Geolife [4], constituée de mesures GPS et de comparer ses performances à celles obtenues avec une approche traditionnelle.

Ce travail est issu d'une action mutualisée entre deux Instituts Carnot : l'institut Carnot IFPEN Transports Energie et l'Institut Carnot Leti.

2 Etat de l'art

2.1 La base Geolife

La base Geolife, construite sur la période 2007-2012 et impliquant 182 utilisateurs, est constituée de trajectoires GPS. Une trajectoire GPS est une suite de points horodatés, chaque point contenant les informations de latitude, longitude et altitude. Une ***trajectoire*** est divisée en ***trajets*** (« trips ») si l'intervalle de temps entre 2 points est supérieur à un seuil prédéfini (typiquement 20 min). Un ***trajet*** peut être multimodal, c'est-à-dire composé d'une suite de trajets monomodaux (« ***triplegs*** ») de durée variable. Ces trajets monomodaux sont à nouveau subdivisés en ***segments*** dont le nombre

de points ***M*** est fixe (typiquement entre 100 et 1000 points) et correspond à la longueur des signaux en entrée du réseau profond.

La base contient 17621 trajectoires pour une distance et durée totales respectives d'environ 1.3 millions de km et 50000 heures.

2.2 Les approches

Les publication basées sur les approches traditionnelles sont assez nombreuses [1] [2] [3] contrairement à celles mettant en œuvre un apprentissage profond pour lesquelles nous n'avons relevé que peu d'études [5] [6] [7] [8].

Etant donné un trajet monomodal, le problème à résoudre ici est d'estimer le mode de transport.

Le problème diffère selon le nombre de modes considéré. Parmi les 11 modes présents dans la base, 4 sont peu représentés ('moto', 'bateau', 'course', 'avion') et sont souvent exclus. Sur les 7 modes restants, les classes 'voiture' et 'taxi' sont souvent confondues par les modèles et sont donc fusionnées la plupart du temps. [7] fusionnent également 'métro' et 'train', bien que ces 2 modes se discriminent assez bien. Enfin, [1] [2] simplifient encore le problème en ne traitant que 4 modes. Nous avons considéré le problème à 6 modes pour lequel les classes 'voiture' et 'taxi' sont fusionnées.

A partir des points GPS, une 1ère étape, commune à la majorité des approches, traditionnelles ou non, consiste, à construire des caractéristiques ponctuelles ***CP***, p. ex., la distance entre 2 points consécutifs (*dL*), la vitesse, l'accélération (*acc*), le jerk (dérivée de *acc*), l'orientation par rapport au Nord, la différence de l'orientation entre 2 points consécutifs (*deltaOrien*), la dérivée de l'orientation (*tauxOrien*). Ensuite, les approches diffèrent : les traditionnelles construisent explicitement des caractéristiques globales au niveau du trajet ***CT***, p. ex. certains percentiles de la vitesse, de l'accélération, la fréquence des arrêts, des virages, … alors que les approches par apprentissage profond utilisent directement en entrée du réseau les ***CP***.

[5] et [6] sont 2 approches qui partagent plusieurs points communs : elles appliquent d'une part un AutoEncoder (***AE***) sur les données d'entrée pour extraire des caractéristiques profondes, et d'autres part elles calculent des caractéristiques globales ***CT*** comme dans les approches traditionnelles. Leurs meilleurs résultats respectivement une justesse (i.e. la proportion d'exemples correctement classés) de 67.9% et 74.1%, sont obtenus quand elles fusionnent ces 2 ensembles de caractéristiques.

[7] décrit des travaux les plus proches de notre approche. Il calcule 4 ***CP*** : vitesse, *acc*, jerk et *deltaOrien* et construit un réseau convolutionnel. La performance annoncée, 82.3% en justesse pour le meilleur réseau, est probablement largement surestimée à cause de la façon de partitionner la base de données (cf. 3.3).

[8] exploite la quantité importante de données non labélisées de la base en réalisant un apprentissage conjoint d'un ***AE*** convolutionnel sur l'ensemble des données et d'un réseau convolutionnel sur les données labélisées. Les performances en justesse sont de 76.8%, l'exploitation des données non labélisées faisant gagner 2.7% par rapport à l'apprentissage d'un réseau convolutionnel basé uniquement sur les données labélisées.

3 Méthode proposée

Nous décrivons ici les principales étapes de nettoyage de la base, prétraitements appliqués au signaux et évaluation des performances de classification.

3.1 Nettoyage de la base

La 1ère étape consiste à mettre en correspondance les données GPS avec les annotations disponibles. Cela restreint la base à 69 utilisateurs. Un nombre conséquent de points (10%) présente un horodatage identique avec le point précédent. Ils sont supprimés.

Une autre difficulté concerne la présence de points GPS aberrants dus à une faible sensibilité des puces GPS utilisées et/ou à la présence de bâtiments en milieu urbain générant des réflexions parasites. Nous avons supprimé ces points (~2% du total) en appliquant des seuils liés à la physique des modes de transport sur les signaux 'vitesse' et valeur absolue de *'acc'* pour chaque mode de transport (cf. Tableau 1).

Tableau 1 : Seuils et nombre d'instances (triplegs) après prétraitements par classe

Classe	marche	vélo	bus	voiture&taxi	métro	train
Seuil vitesse(m/s)	4	20	36	36	36	100
Seuil accél. (m/s²)	5	5	10	10	10	5
Nombre d'inst.	4517	2129	1731	1459	632	200

Nous avons également supprimé les triplegs trop courts, i.e. dont le nombre de points GPS est inférieur à un seuil prédéterminé (typiquement 10 points).

Au final, la base nettoyée comporte 65 utilisateurs, 10668 trajets monomodaux, pour une distance et une durée totales respectivement d'environ 103000 km et 4600 heures.

3.2 Prétraitements

Le premier prétraitement consiste à calculer un ensemble de caractéristiques ponctuelles, tel que décrit en 2.2. Comme les données présentent une période d'échantillonnage variable nous avons interpolé linéairement les signaux 'vitesse' et 'orientation' avec un pas constant (égal à la médiane de la période d'échantillonnage soit 2 s).

Les caractéristiques correspondant à des grandeurs physiques différentes, chaque caractéristique est normalisée. Deux types de normalisation ont été testées : la normalisation 'MinMax', qui transforme linéairement les données à partir des valeurs minimum (transformée en 0) et maximum (transformée en 1) et une normalisation 'p-robuste', p définissant un percentile entre 0 et 100, qui associe aux percentiles p et 100-p les valeurs 0 et 1.

Enfin, pour chaque ***tripleg***, les caractéristiques associées sont segmentées en ***segments*** de ***M*** points, le dernier segment étant complété par des 0.

3.3 Evaluation des performances

Dans le cas de bases de données de petite taille, l'évaluation des performances d'un modèle est généralement réalisée par validation croisée sur un nombre K (typiquement 5 à 10) de sous-ensembles. L'inconvénient de cette approche est qu'elle nécessite au minimum K apprentissages.

Si la base de données est plus conséquente, on peut se contenter d'une partition en 3 sous-ensembles : apprentissage *A*, validation *V* et test *T*. *A* est utilisé pour apprendre les paramètres du modèle, *V* permet d'optimiser le choix des hyper-paramètres, enfin *T* donne une estimation des performances du modèle retenu. Ces 3 sous-ensembles doivent être indépendants ; dans le cas contraire, l'estimation des performances sera biaisée de façon optimiste.

Dans le cas de données temporelles, l'autocorrélation des signaux peut être problématique si la partition des données est réalisée de façon aléatoire sur les segments.

La figure 1 présente un exemple de signaux temporels pour un trajet en train. On note que pour les grandeurs vitesse et orientation, la corrélation (figure 2) demeure importante (>0.2) pour des décalages importants (jusqu'à 750 points). Ceci signifie que si on segmente ce trajet avec un nombre de points ***M*** petit comparativement à l'autocorrélation (e.g. ***M*** =200), alors ces segments ne seront pas indépendants. Si la partition est réalisée par la suite, alors l'estimation des performances sera biaisée. Pour éviter ce risque, nous réalisons d'abord l'opération de partition sur les triplegs, puis l'opération de segmentation. De cette façon, les segments issus d'un même tripleg restent dans le même sous ensemble.

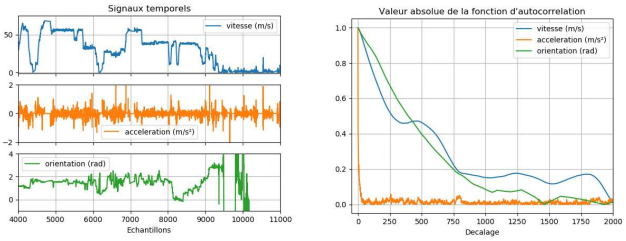


Figure 1 (g): un exemple de signaux temporels pour un trajet en train – Figure 2 (d) : les fonctions d'autocorrélation associées

4 Résultats et discussion

Dans cette partie, nous présentons les résultats obtenus par apprentissage profond (4.3). Pour avoir une base de comparaison, nous avons mis en œuvre une approche traditionnelle (4.2). Les traitements communs aux 2 approches sont présentés dans (4.1).

4.1 Traitement commun

Les données ont été normalisées selon une normalisation '5-robuste'. Les proportions respectives des 3 sous-ensembles *A*, *V* et *T* sont 64, 16 et 20%. Pour prendre en compte le déséquilibre de la base (cf. Tableau 1), un poids, inversement proportionnel au nombre d'exemples de chaque classe, a été affecté à chaque exemple de la base. Chaque apprentissage est réalisé 3 fois, en modifiant à chaque fois les différentes graines aléatoires (pilotant la partition et l'initialisation des poids).

Le critère de performance retenu est le score F1 moyenné sur les 6 modes et calculé sur la base de test. Les résultats seront présentés sous la forme d'une moyenne ± écart type, calculés sur les 3 apprentissages. Pour pouvoir se comparer à l'état de l'art, on calcule également une justesse, bien que cela soit moins pertinent dans le cas d'une base de données non équilibrée.

La mise en œuvre est réalisée en Python à l'aide des libraires Scikit-Learn et Keras.

4.2 Approche classique

En s'inspirant de [1], 7 descripteurs ont été construits manuellement : les percentiles 5 et 95 de l'accélération, la vitesse moyenne (ratio de la distance sur la durée du trajet), le percentile 95 de la vitesse, la moyenne de la valeur absolue de la dérivée de l'orientation, la distance moyenne entre 2 arrêts et entre 2 virages.

Différents couples (classifieurs, hyper-paramètres) ont été testés : classification naïve bayésienne, arbre de décision, machine à vecteurs supports et perceptron multicouche. Les meilleurs résultats, F1 = 75.4±0.5%, ont été obtenus pour un réseau de neurones à 1 couche cachée avec 256 neurones.

4.3 Apprentissage profond

Différents jeux d'hyper-paramètres ont été testés (voir 4.4). Les meilleurs résultats obtenus, F1=78.6±1.0%, sont supérieurs de 3.2% à l'approche traditionnelle. La justesse de 81.9±0.3% montre que l'on améliore les performances de 5% par rapport à l'état de l'art. Les figures 3 et 4 présentent la matrice de confusion pour un des 3 apprentissages, respectivement pour l'approche classique et l'apprentissage profond. Dans les 2 cas, 'marche' et 'vélo' sont bien estimés (F1>79%), et la principale confusion concerne les classes 'voiture' et 'bus'.

		Matrice de confusion						Performance par classe			
		velo	bus	voiture	metro	train	marche	Support	precision	recall	f1-score
Vérité	velo	300	16	1	5	0	24	346	0.76	0.87	0.81
	bus	43	292	52	21	4	14	426	0.79	0.69	0.73
	voiture	14	60	162	35	15	6	292	0.68	0.55	0.61
	metro	3	1	20	83	7	12	126	0.56	0.66	0.61
	train	0	0	2	3	35	0	40	0.57	0.88	0.69
	marche	34	0	0	0	0	870	904	0.94	0.96	0.95
		Prediction									

Figure 3 : Résultats de l'approche classique sur la base de test

		Matrice de confusion						Performance par classe			
		velo	bus	voiture	metro	train	marche	Support	precision	recall	f1-score
Vérité	velo	482	39	13	18	0	65	617	0.81	0.78	0.79
	bus	48	675	172	43	4	54	996	0.81	0.68	0.74
	voiture	51	104	463	28	37	43	726	0.69	0.64	0.66
	metro	10	16	10	167	3	27	233	0.63	0.72	0.67
	train	1	2	9	8	351	5	376	0.89	0.93	0.91
	marche	5	0	0	1	0	1540	1546	0.89	1.00	0.94
		Prediction									

Figure 4 : Résultats de l'apprentissage profond sur la base de test

Les hyper-paramètres associés à ce réseau sont :

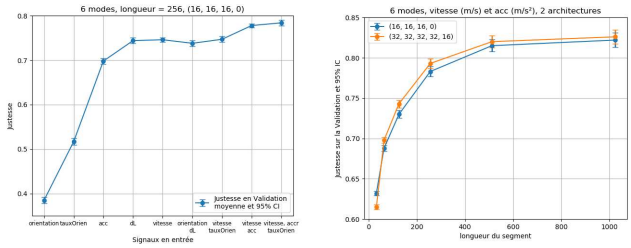
- 'vitesse' et 'accélération' en entrée du réseau, avec une durée correspondante à ***M*** = 512 échantillons temporels,
- Une architecture à 4 couches de convolution (32 filtres, filtre de longueur 3), un MaxPooling à 2 échantillons, une activation 'ReLU', une couche dense à 16 neurones.
- Un algorithme d'optimisation 'AdaDelta' avec les paramètres par défaut.

- Un terme de régularisation de 0.001, un 'Dropout' de 0.2 appliqué après chaque couche dense ou de MaxPooling.

4.4 Sensibilité

Le choix du réseau final présenté ci-dessus résulte d'une étude de la sensibilité des résultats (en terme de justesse sur la base de validation) à différents hyper-paramètres. Une architecture sera décrite par un 'n-uplet' ; les n-1 1ers nombres faisant référence aux nombres de filtres de chaque couche de convolution, le dernier à la couche dense (0=aucune couche dense)

La figure 5 présente l'influence du choix des signaux sur les performances pour une architecture (16,16,16,0) et une longueur (***M*** =256) de données. On observe que si on considère un unique signal en entrée, c'est la vitesse (ou *dL* qui lui est proportionnelle) qui conduit à la meilleure performance. Rajouter à la vitesse une information d'orientation n'améliore pas les résultats ; par contre, rajouter l'accélération améliore les performances d'environ 2%. La figure 6 montre que la valeur de ***M*** influe fortement sur les résultats et que notre choix de 512 (correspondant à un trajet d'environ 17 minutes) semble un bon compromis entre performances, taille et latence du réseau.



Influence du choix des signaux (Figure 5, g.) et de leur longueur (Figure 6 à d.) sur la justesse de la base de validation

Le type de normalisation influe grandement sur les performances et nous avons observé un gain de 7% entre une normalisation 'MinMax' et une normalisation '5-robuste'. Ceci peut s'expliquer par le fait que la valeur maximum de la vitesse (~100 m/s) est bien supérieure au 95ème percentile (~22 m/s).

Nous avons étudié l'influence de l'architecture sur les performances. Nous avons testé de 1 à 4 couches de convolution, pour 2 jeux de signaux en entrée ('vitesse' seule et 'vitesse'+'*acc*'), ainsi que la présence ou non d'une couche dense. Les principaux résultats sont présentés sur les figures 7 et 8 et montrent qu'il y a un optimum autour de 4 couches de convolution et de 30000 paramètres.

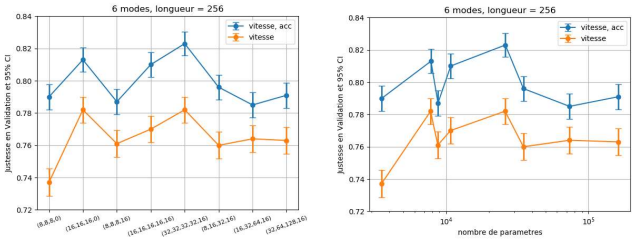


Figure 7 (g) : Influence de l'architecture sur la justesse en validation. Figure 8 (d) : les mêmes résultats en fonction du nombre de paramètres de chaque architecture.

L'influence de la façon de partitionner les données (cf. 3.3) a été évaluée pour un signal 'vitesse' en entrée de taille ***M*** = 256 et une architecture (32, 32, 32, 32, 16) : la performance sur la base de test *T* passe de F1= 70.6±1.6% (partition puis segmentation) à 75.2±1.4% (segmentation puis partition). La surestimation des performances peut donc atteindre 4.6%.

Enfin, la longueur de filtre a été étudiée pour les valeurs 3, 7, 11 et 15. Elle influe peu sur les résultats, les meilleurs étant obtenus pour la plus petite valeur (3).

5 Conclusions

Sur la base de données publique Geolife, nous avons développé et optimisé un réseau convolutionnel profond.

Les performances obtenues, de l'ordre de 79% (score F1) et 82% (justesse) sont supérieures de 3% à celles obtenues par une approche traditionnelle et de 5% aux approches similaires décrites dans l'état de l'art.

Nos principales contributions ont été de montrer que :

1) Les signaux les plus pertinents sont la vitesse et l'accélération,

2) La durée des signaux en entrée du réseau doit être suffisante, de l'ordre de 17 min.

3) Le signal 'vitesse' présente une autocorrélation importante, de l'ordre de plusieurs minutes, qui peut conduire à une surestimation des performances du modèle, de l'ordre de 5%, si la partition des données n'en tient pas compte.

Les perspectives sont d'interpréter les différents filtres obtenus et de prendre en compte les données non labélisées de la base.

6 Références

[1] Y. Zheng, Y. Chen, Q. Li, X. Xie, et W.-Y. Ma, « Understanding Transportation Modes Based on GPS Data for Web Applications », *ACM Trans Web*, vol. 4, n° 1, p. 1:1–1:36, janv. 2010.

[2] H. Mäenpää, A. Lobov, et J. L. Martinez Lastra, « Travel mode estimation for multi-modal journey planner », *Transp. Res. Part C Emerg. Technol.*, vol. 82, n° Supplement C, p. 273-289, sept. 2017.

[3] Z. Xiao, « Identifying Different Transportation Modes from Trajectory Data Using Tree-Based Ensemble Classifiers », 2017.

[4] Zheng, Yu, « Geolife GPS trajectory dataset – User Guide », version 1.3, août 2012.

[5] Y. Endo, H. Toda, K. Nishida, et J. Ikedo, « Classifying spatial trajectories using representation learning », *Int. J. Data Sci. Anal.*, vol. 2, n° 3-4, p. 107-117, déc. 2016.

[6] H. Wang, G. Liu, J. Duan, et L. Zhang, « Detecting Transportation Modes Using Deep Neural Network », *IEICE Trans. Inf. Syst.*, vol. E100.D, n° 5, p. 1132-1135, 2017.

[7] S. Dabiri et K. Heaslip, « Inferring transportation modes from GPS trajectories using a convolutional neural network », *Transp. Res. Part C Emerg. Technol.*, vol. 86, p. 360-371, janv. 2018.

[8] S. Dabiri, « Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using GPS Trajectory Data », 2019.