

Prédiction de séries temporelles multi-variées stationnaires: modélisation du contexte pour l'analyse des données de transports

Valentin GUIGUET¹, Perrine CRIBIER-DELANDE^{1,2}, Nicolas BASKIOTIS¹, Vincent GUIGUE¹

¹Laboratoire D'informatique de Paris 6, Sorbonne-Université
4 place Jussieu F-75252 PARIS cedex 05

²Renault, Direction DEA-IR
Technocentre, 1 avenue du Golf 78084 GUYANCOURT CEDEX, France
prenom.nom@lip6.fr

Résumé –

Cet article décrit un nouvel algorithme prédictif dédié aux séries temporelles multi-variées qui se base non seulement sur les relations entre les valeurs passées et futures mais qui apprend aussi une représentation de certaines caractéristiques externes comme le contexte spatio-temporel. Nous démontrons l'intérêt de cette approche sur des données issues d'un système de transports en commun : notre algorithme d'apprentissage de représentation (qui utilise des réseaux de neurones récurrents) permet de dépasser significativement les méthodes de référence pour la prédiction de fréquentation tout en modélisant élégamment le contexte spatio-temporel des séries analysées.

Abstract –

This article presents a new predictive model for multivariate time series that not only makes use of the relations between past and future values but also learns the representation of some exogenous features such as spatio-temporal context. We demonstrate the advantages of our model by applying it to the forecast of smart cards tap-in logs in the Parisian subway network: our model (that uses Recurrent Neural Network to learn representations) outperforms the baseline for forecasting while also elegantly learning the spatio-temporal context.

1 Introduction

Nous nous intéressons à la prédiction de séries temporelles multi-variées dans le cadre de l'analyse du réseau de transports en commun d'Île-de-France. La mobilité urbaine est un domaine qui présente une vaste littérature hétéroclite allant de la prédiction de flux à l'aménagement du territoire en passant par la modélisation des comportements ; les études abordent ces problématiques à différentes échelles (ensemble du territoire, sous-zones, individus. . .). La disponibilité, depuis le début des années 2000, de nombreuses données décrivant la mobilité urbaine offre un nouveau regard sur la dynamique des populations. Plusieurs types de données permettent ces études : les logs des usagers dans les réseaux de transport en commun [1], [2], [3], les mouvements enregistrés par les GPS des flottes de taxis [4], les données de localisation des bornes de téléphonie mobile voire les systèmes de paiement par carte bancaire [5].

Nous nous concentrons sur les données AFC (Automated Fare Collection) issues de *Île-de-France Mobilités* qui décrivent le nombre d'usagers entrant dans les 300 stations de métro du réseau au cours du temps. Ces séries multi-variées sont stationnaires, les journées présentant des régularités très fortes pour une station donnée. Chaque journée sera considérée comme une série indépendante des autres. La tâche que nous abordons est la prédiction de la fréquentation dans les stations du réseau

à $t + \delta t$, à partir des données de la série temporelle de 0 à t . Les réseaux de neurones récurrents (RNN) sont un choix naturel pour faire de la prédiction car ils encodent dans leur couche cachée une représentation du passé sur un horizon assez long qui permet de prédire l'état suivant de manière efficace [6].

Notre contribution consiste :

- à proposer différentes architectures de RNN permettant d'encoder le contexte des séries temporelles (jour de la semaine, station concernée),
- à comparer ces modèles à des modèles de référence simples et efficaces sur différents horizons de prédiction.

Après avoir décrit les données et les différents modèles, nous comparerons et analyserons les performances obtenues. Nous montrerons que nos propositions permettent d'améliorer significativement la prédiction de la fréquentation dans le réseau de transport, en particulier dans le cas difficile des prédictions à long terme (plusieurs heures).

2 Modèles

Nos données sont regroupées dans un tenseur \mathbf{X} de taille $N \times S \times T$ où N est le nombre de jours (77), S le nombre de stations (297) et T le nombre de quart d'heure dans une journée (96). Ainsi, une observation pour une station s est un

vecteur de taille T dénoté \mathbf{x}^s et nous disposons de N observations indépendantes pour cette station. Pour chaque signal, nous connaissons donc la station associée mais aussi le jour de la semaine concerné. Le problème de prédiction consiste à prédire x_{t+1}^s (ou $x_{t+\delta t}^s$) à partir des pas de temps précédents.

Modèle de référence Une des particularités des séries temporelles de transport est leur stationnarité et leur dépendance très forte au contexte. D’abord, la fréquentation dépend beaucoup de la station (contexte spatial). Ensuite, il existe aussi une dépendance au jour de la semaine considéré (contexte temporel). Cela nous permet de créer une référence simple en calculant une moyenne de la fréquentation à chaque pas de temps par station et par jour de la semaine.

Ce modèle est donc un tenseur de taille $S \times T \times 7$. Il donne un comportement moyen et est statique : il ne permet pas de s’adapter à d’éventuelles fluctuations.

Réseaux de neurones récurrents (RNN) Nous avons utilisé trois types de RNN différents : le RNN classique, les LSTM (Long-Short Term Memory) et les GRU (Gated Recurrent Unit). Ces deux derniers servent à palier les difficultés liées à la disparition des gradients, un problème inhérent aux RNN.

Nous proposons trois architectures qui diffèrent dans la manière dont elles encodent le contexte spatial. Chaque architecture est composée d’un encodeur qui encode la séquence d’observations dans son état caché. Cet état est ensuite fourni à un décodeur qui fait une prédiction sur l’état suivant. Le décodeur est un réseau avec une couche cachée.

Dans un premier temps, nous avons considéré chaque station indépendamment et entraîné S encodeurs distincts (Figure 1 (a)), qui considèrent des séries uni-variées. Nous faisons donc l’hypothèse que les stations ont des dynamiques différentes et nécessitent donc d’être modélisées séparément. Au temps t , pour la station s , la séquence d’observations $x_{0:t}^s$ est encodée par le RNN correspondant E^s . La couche cachée de ce RNN h_t^s est ensuite donnée au décodeur D^s qui génère une prédiction x_{t+1}^s .

Notre deuxième proposition : le modèle multi-varié (illustré en Figure 1 (b)) considère qu’à chaque pas de temps, les fréquentations des différentes stations ne sont pas indépendantes. Nous utilisons les observations de tout le réseau pour prédire la fréquentation de chaque station. Notre entrée est donc une série temporelle multi-variée de taille S et notre architecture prédit directement $\mathbf{x}_{t+1} \in \mathbb{R}^S$.

Le troisième modèle, qui contient du contexte spatial, est une approche hybride des deux premiers. Comme dans le modèle uni-varié, nous avons $N \times S$ exemples de taille T . En revanche, il n’y a qu’un seul couple d’encodeur/décodeur (E, D) partagé par toutes les stations, comme dans le modèle multi-varié. Nous allons apprendre une représentation explicite de chaque station. Chaque observation sera concaténée avec la représentation de sa station de rattachement puis le tout sera mis en entrée de l’encodeur. Nous apprenons donc une représentation z_s correspondant au contexte spatial de taille λ_s (Figure 1 (c)). C’est

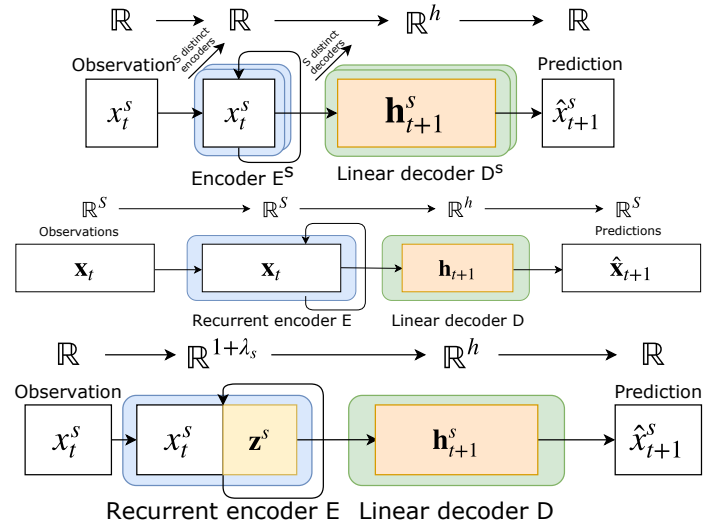


FIGURE 1 – (a) RNN uni-varié (S modèles indépendants), (b) RNN multi-varié, (c) RNN unique avec encodage du contexte spatial.

une manière élégante de conserver un modèle compact tout en distinguant les différentes stations ; l’encodeur devra tirer parti de la description des stations tandis que le décodeur bénéficiera d’un apprentissage sur les signaux de l’ensemble du réseau.

Nous avons encodé de manière similaire le contexte temporel correspondant au jour de la semaine dans une représentation vectorielle $\mathbf{z}_d \in \mathbb{R}^{\lambda_d}$, $d = 1..7$. Enfin, nous avons aussi encodé la position temporelle (le quart d’heure) dans $\mathbf{z}_t \in \mathbb{R}^{\lambda_t}$, $t = 1..96$. Ces deux représentations permettent d’encoder toutes les informations dont dispose le modèle de référence pour que le RNN se concentre sur la modélisation des déviations.

3 Expériences

Nous travaillons sur des données de transport fournies par Île-de-France Mobilités. Le jeu de données contient 256M de logs (horodatage, station) de validation de cartes magnétiques entre Octobre et Décembre 2015 pour le réseau de métro (300 stations). Sur les 300 stations, 3 étaient en travaux pendant tout ou une partie de la période considérée.¹ L’horodatage est discrétisé par plage de 15 minutes.

Les jours sont séparés en ensembles d’entraînement, de validation et de test avec une proportion 60/10/30. Il y a donc 45 jours dans l’ensemble d’entraînement, 8 dans celui de validation et 24 pour l’ensemble de test.

Mise à l’échelle Le réseau de métro parisien contient quelques stations principales très fréquentées et beaucoup de stations plus petites. De fait, ces différences d’échelle compliquent

1. Sur les 92 jours de nos données, 15 jours "anormaux" ont été retirés. Le 11 Novembre car c’est un jour férié, le 14 Novembre car c’était le lendemain des attaques de Paris, les 29 et 30 Novembre car les transports étaient gratuits pour cause de COP21 et enfin les vacances de Noël (du 21 Décembre à la fin de l’année).

notre tâche, en particulier pour l’architecture multi-variée. Nous avons donc décidé de normaliser les données station par station. Nous commençons par calculer le 99.9ème percentile de chaque station et remplaçons toute valeur supérieure ou égale par celle-ci. Ensuite, les valeurs sont mises à l’échelle entre -1 et 1 via min-max, station par station pour éviter que les stations principales n’écrasent les plus petites. Ces calculs sont fait sur l’ensemble d’entraînement puis les valeurs sont appliquées aux autres ensembles.

Détails pratiques sur les modèles Nous avons utilisé des RNN classiques ainsi que des Gated Recurrent Units pour l’encodeur. Les modèles sont entraînés en pytorch sur des GPUs en utilisant l’optimiseur Adam avec un pas d’apprentissage de 0.0001 et la fonction de coût est une MSE (Mean Square Error).

Les résultats sont donnés en Root Mean Square Error (RMSE) une fois la mise à l’échelle inversée :

$$RMSE(x, \hat{x}) = \sqrt{\frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{n}}$$

Les hyper-paramètres choisis sont présentés ci-dessous et nous utilisons comme taille de représentation : $\lambda_s = 80$, $\lambda_d = 4$ and $\lambda_t = 30$. Chaque expérience a été lancée 5 fois avec une initialisation de l’aléatoire différente pour nous permettre de calculer la variance de l’erreur.

	Batch size	Layers	Hidden size
Multi-varié	2	1	400
Uni-varié	1	1	100
Spatial	128	2	200

Nous présenterons dans les sections 3.1 et 3.2 les résultats pour les prédictions à $t + 1$ et dans la section 3.3 les résultats pour les horizons plus lointains.

3.1 Résultats

La comparaison des résultats selon les différentes architectures est présentée en Table 1. Comme attendu, la RMSE est plus élevée pour les modèles utilisant des RNN basiques au lieu de GRU ou de LSTM. Les GRU et les LSTM obtenant des résultats sensiblement identiques, nous ne présenterons que les résultats des GRU. Les modèles contextuel et uni-varié sont plus performants que le modèle multi-varié ce qui montre le gain de l’intégration du contexte spatial. La figure 2 montre les prédictions des différents modèles pour le 4 Novembre, afin de vérifier que nos modèles ont appris plus que le comportement moyen. En effet, ce jour là une anomalie est survenue, ne permettant pas à la référence d’avoir de bonnes prédictions. En revanche, les modèles appris se sont adaptés à la fréquentation inhabituellement basse. Cela montre que les modèles ont bien capturé la dynamique des séries temporelles et sont robustes aux situations inconnues. De plus, nous constatons que le modèle contextuel spatial est plus précis que l’uni-varié. Cette représentation remplace donc avantageusement les

multiples RNN nécessaires de l’architecture uni-variée, en utilisant beaucoup moins de paramètres : le modèle contextuel spatial est composé de 434 961 paramètres face aux 9 207 292 paramètres du modèle basé sur 297 GRU (≈ 20 fois moins).

TABLE 1 – RMSE de la prédiction des fréquentations sur des fenêtres de 15 minutes pour chacune de nos architectures et comparaison avec la référence.

	RNN	GRU	Référence
Multi-varié	28.31 ± 0.09	27.83 ± 0.12	
Uni-varié	26.15 ± 0.08	26.73 ± 0.18	31.98
Contexte spatial	24.98 ± 0.05	24.96 ± 0.05	

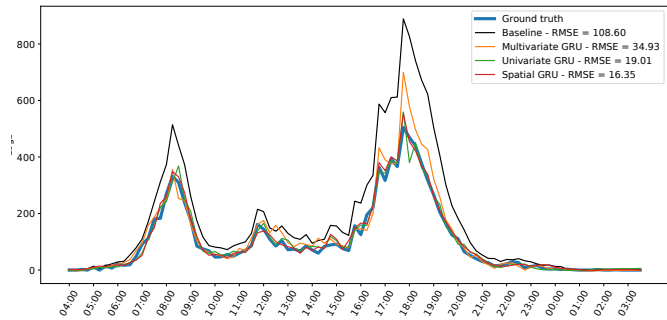


FIGURE 2 – Prédictions de la fréquentation de la station Pont de Levallois-Becon le mercredi 4 Novembre. Il y a eu une fluctuation ce jour là, ce qui rend les prédictions de la référence imprécises alors que les RNN se sont adaptés au changement.

3.2 Améliorations dues au contexte temporel

La représentation du contexte temporel (comme celle du contexte spatial) nous permet d’encoder toutes les informations dont dispose le modèle de référence dans les représentations fournies au RNN contextuel qui apprendra alors la dynamique des écarts au modèle moyen. Nous avons appris des représentations des 7 jours de la semaine et des 96 quarts d’heure d’une journée en étendant simplement chacune des trois architectures précédentes. Les résultats sont présentés dans la Table 2.

L’ajout de contexte temporel est bénéfique pour tous les modèles, à l’exception de la représentation du jour pour les modèles uni-varié et multi-varié en GRU. Le modèle qui bénéficie le plus de l’ajout du contexte temporel est le modèle hybride, qui apprend déjà une représentation du contexte spatial.

3.3 Prédictions à un horizon plus lointain

Les expériences précédentes se contentaient de prédire la valeur de fréquentation au pas de temps suivant. Nous voudrions étudier la capacité de prédiction des modèles sur des horizons supérieurs au quart d’heure. Pour cela, nous ré-injectons les prédictions dans le RNN afin de prédire les pas de temps plus lointain. Les erreurs de prédiction du modèle vont se propager et dégrader la qualité des prédictions suivantes. La Fi-

TABLE 2 – RMSE des prédictions de fréquentation en utilisant des représentations du contexte temporel et des GRU

	Sans contexte	Jour	Heure	Jour et heure	Référence
Multi-varié	27.83 ± 0.12	27.99 ± 0.11	27.36 ± 0.12	27.24 ± 0.07	
Uni-varié	26.73 ± 0.18	25.16 ± 0.09	26.76 ± 0.14	24.93 ± 0.12	31.98
Spatial	24.96 ± 0.05	24.73 ± 0.07	24.96 ± 0.08	24.77 ± 0.13	

Figure 3 illustre cette évolution. Les prédictions des modèles sans contexte sont dépassées par la référence au bout d'environ une heure (4 prédictions). La prise en compte des contextes spatial et temporel améliore nettement la stabilité et permet de doubler l'horizon de temps auquel les RNN restent compétitifs.

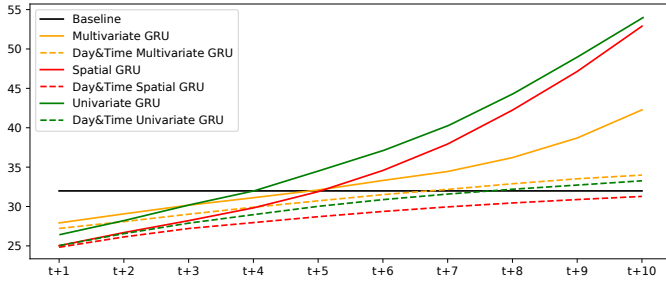


FIGURE 3 – RMSE en fonction de l'horizon temporel avec réinjection des prédictions précédentes. Les valeurs à t+1 correspondent aux tableaux précédents. Les pointillés correspondent aux modèles utilisant les représentations temporelles.

Pour confirmer que le contexte temporel est particulièrement utile lorsque nous prédisons plus loin dans le futur, nous proposons une autre expérience. Nous nous sommes limités aux p premières observations d'une séquence pour initialiser la couche cachée de l'encodeur avant de prédire le reste de la journée. Les résultats de cette expérience sont présentés en Figure 4 pour $p = 16$ (les prédictions commençant donc à 8h). La première figure (4a), montre la moyenne de la différence en RMSE entre la référence et les trois modèles récurrents. Nous constatons que les performances du modèle multi-varié de base se détériorent à mesure que la journée avance, en particulier durant les heures de pointe. Les modèles qui prennent en compte le contexte temporel, en revanche, convergent vers le modèle moyen. En effet, sur le long terme, prédire un comportement moyen est la meilleure manière de minimiser la MSE. De plus, la figure 3 montre que les prédictions sont même meilleures jusqu'à 1 heure après la dernière valeur réelle observée.

Nous avons ensuite appliqué ce protocole à l'exemple anormal présenté plus haut. Dans ce cas particulier, la référence n'est pas un bon estimateur. Nous constatons sur la figure 4b que les modèles appris ont détecté dès la 4ème heure que le trafic était perturbé et se sont adaptés. Même dans une situation anormale, les modèles sont compétitifs sur les prédictions long terme.

4 Conclusion

L'ajout de représentations contextuelles spatio-temporelles permet l'amélioration des prédictions des séries temporelles.

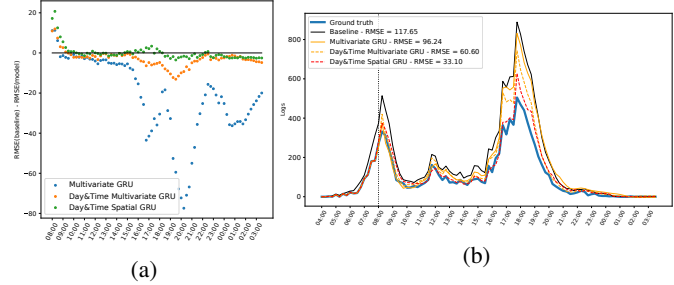


FIGURE 4 – Prédiction sur l'ensemble de test, avec réinjection des prédictions précédentes à partir du 16ème pas de temps. En (a) la moyenne de la différence en RMSE entre la référence et nos modèles. En (b) la prédiction de fréquentation pour le 4 Novembre à la station Pont de Levallois-Becon

En particulier, le contexte spatial permet d'améliorer les prédictions à court terme des RNN en les rendant plus robustes et moins sensibles aux anomalies que la référence. Le contexte temporel permet de garder cette robustesse sur des horizons plus lointains.

Références

- [1] Catherine Morency, Martin Trépanier, and Bruno Agard. Measuring transit use variability with smart-card data. *14(3)* :193–203, 2007.
- [2] Irina Ceapa, Chris Smith, and Licia Capra. Avoiding the crowds : understanding tube station congestion patterns from trip data. pages 134–141. ACM, 2012.
- [3] Sui Tao, David Rohde, and Jonathan Corcoran. Examining the spatial-temporal dynamics of bus passenger travel behaviour using smart card data and the flow-comap. *41* :21–36, 2014.
- [4] Lin Xu, Yang Yue, and Qingquan Li. Identifying urban traffic congestion pattern from historical floating car data. *Procedia - Social and Behavioral Sciences*, 96 :2084 – 2095, 2013.
- [5] Thomas Louail, Maxime Lenormand, Juan Murillo Arias, and José J Ramasco. Crowdsourcing the robin hood effect in cities. *Applied network science*, 2(1) :11, 2017.
- [6] Florian Toqué, Mostepha Khouadjia, Etienne Come, Martin Trepanier, and Latifa Oukhellou. Short & long term forecasting of multimodal transport passenger flows with machine learning methods. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pages 560–566. IEEE, 2017.