Motion ResNet : An efficient data imputation method for spatio-temporal series

1st Mathieu Crilout
LIP6, Sorbonne University
Paris, France
mathieu.crilout@lip6.fr

1st Nicolas Baskiotis
LIP6, Sorbonne University
Paris, France
nicolas.baskiotis@lip6.fr

1st Vincent Guigue
LIP6, Sorbonne University
Paris, France
vincent.guigue@lip6.fr

1st Lynda Said Lhadj
LIP6, Sorbonne University
Paris, France
lynda.said@lip6.fr

***Abstract*—Fleet monitoring system have limitations to save GPS positions across time and can also collect corrupted data. Data imputation is the task consisting to infer missing data points and can solve both problems by augmenting artificially the GPS resolution across time and infer corrupted data points. Several neural net architectures have been used successfully to solve different tasks of data imputation. Here we propose a novel approach using the recent discovered link between ResNet and Integration to relay on equations of motion and to take account of spatial dependencies. We show that it outperforms classical interpolation methods on various data imputation tasks and that can easily incorporate any other priors.**

I. Introduction

Nowadays, more and more data are available of many kinds, due to the large deployment of various sensors measuring different phenomenons. Mobility data takes part of this tendency, and more specifically fleet tracking systems, which are growing rapidly and monitoring bigger and bigger fleets, aggregating trajectories of many vehicles. In this process, some problems occurs, as bandwidth limitations, random corruptions or broken devices, which leads to partial corrupted data, missing data, sub-sampled data, and sometimes various sampling rates across time.

A fleet monitoring system works mainly with an embedded system for each monitored vehicle. This system contains a GPS module to assess vehicle's positions and instantaneous speeds across time. It also contains a GSM module to transmit registered information from the GPS module to a central database. The main limitation is that the central database has a limited bandwidth and if it works as real time monitoring, it cannot store all information transmitted and leads to limited trajectories resolution.

Yet, trajectories resolution is directly linked with data quality and if we want to characterize driving behaviour, or evaluate accurate traffic evolution, we need a measure frequency rate of 1 or more. Thus, data imputation techniques can help to overcome this technical limitations by artificially augmenting frequency rate with inference techniques. Moreover, it can normalizes the frequency over the whole dataset, something mandatory for many models which take fixed time series resolution.

Many models have been used to do data imputation for time series. Simple interpolations or simple statistical inferences were used, also k-Nearest Neighbors, ARMA based models, Markov Chain Monte Carlo models, gaussian processes, or more sophisticate statistical inference methods. Most of them taking part of the data structure they are trying to impute.

In the case of spatio-temporal series, priors over spatial correlation matters. Many of the previous methods use arbitrary spatial correlation window, an assumption that is false in our case since when you follow a street the two opposite directions might have really different traffic behaviour even if they follow the same path, while for some other method it is really not clear to do so, and incorporate spatial information can be challenging.

Neural nets are more and more popular and are used for a broad kind of problems, in data imputation some papers using neural nets claim being the state of the art. Architecture and methods really depend of the data structure of the problem, but it is rather clear that neural nets is a promising method to do data imputation.

Moreover, many papers showed a strong connection between ResNet architecture and equation solver as euler forward method, and differential equation is a pretty natural framework to do spatio-temporal modelling. The approach of using equation system as prior over neural nets architecture has already been used successfully in different applications : weather forecast, physical process modeling, signal reconstruction. Following this two observations, this paper proposes a new architecture of neural net inspired from equations of motion but letting the model loosely recovering the system and incorporating spatial representation to condition each equation of motion with a spatial context.

In this paper we introduce a novel neural net architecture called Motion ResNet with several advantages :

- Our model links spatial and temporal dimensions naturally, overcoming the difficulty to integrate both at the same time.
- It can integrate other information as, for instance, weather data, traffic context, driver identity, and thus can use all the information available.
- It can be used with any type of missing data, random missing data, various sampling rates or to augment resolution artificially.
- It is usable in both Supervised and Unsupervised fashions.
- It has promising result on a real dataset of car's trajectories in real situation.

II. Related Work

A. Residual Networks, Recurrent Neural Networks and Differential Equations

Many papers already linked RN with ODEs and showed a direct correspondence between forward function and specific approximation of differential equations, , , . Thus many papers used this knowledge to build specific neural network inspired from existing differential equations, , , . Since then, a lot of deep neural networks have been used to model complex dynamical systems using knowledge about differential equations as good prior over deep learning architectures , , , .

Let us consider $W_t(X_t)$ a transition function to be fit by a few stacked layers with X_t denoting the input. A residual building block with N layers is defined with the following forward pass :

$$X_{t+1} = X_t + hW_t(X_t) \text{ for t = 0, ..., N-1.} \quad (1)$$

For each layer, a skip connection is used so that the output in the end represents a slightly different version of the input. The transformation being entirely represented with the residual term $hW_t(X_t)$. In fact this method can be seen as a forward Euler discretization with a fixed step size h and N steps. Thus you can see any forward pass of a residual building block as a kind of discretization approximation method to solve ordinary differential equations (ODEs). It can also be interpreted as gradient flows for the Wasserstein metric on the space of probability measures. So you can see your neural network as a mapping function between two distribution : your input data and your output (in our case the initial conditions to the final states of our trajectories). Each residual building block being a gradient flow step.

Recurrent neural network (RNN) models can be seen as a mapping function between sequential dynamical states $x = x_1, .., x_T$. At each time step t, RNN takes the input vector x_t to map it to its next state. In fact in its most generic form it can be seen as the following operator F:

$$x_{t+1} = F(x_t) = \frac{d}{dt}x_t \quad (2)$$

F is a mapping function of any kind and so can be formed with residual building blocks. Residual Recurrent Neural network were first introduced by , and were used in several applications, , , You can think of it as a continuous mapping between temporal state and so is strongly connected with the former vision of ODEs where it represents the evolution of dynamical system. Many works explored classical dynamical systems represented with ODEs and residual recurrent neural networks (RRNN). suggests that RRNN is an exact ODEs solver and so is slightly different than discretization forward solver methods.

B. Spatio Temporal series and data Imputation

Imputing missing values on multivariate time series lead to many interesting works in various practical applications ranging from biology, geoscience or even healthcare, , ,. There are specific works for spatio-temporal time series already but managing to use both spatial prior and temporal is challenging. used successfully deep learning methods to tackle this problem. To infer missing data given incomplete sequential input $x = x_1, ..., x_T$, most of this methods use inferred state $\tilde{x_t}$ if the true state x_t is unknown. In fact you can see this method as a local forward Euler discretization over the missing states since as we said above any RNN can be seen mapping operator between two consecutive state. So in the next section we introduce our model which generalize this method as very promising and versatile way to impute sequencing data.

III. Equations of motion

IV. Motion ResNet

A. Equations of motion

equations of motion

B. Model architecture

Base line = equations of motion First simple RNN with each input being the current inferred state (to compare with equations of motion).

Second RNN with each input being the current inferred state + representation of road portion

Then BI-RNN versions of the two previous ones.

Then all last versions with higher orders mapping transition (speed + acceleration)

V. Experiment

In this section, we validate the efficiency of our proposed model for time series data in different missing rate. The missing time series imputation method is based on the dependencies of past observed data and all known sample variables and intertwined with local spatial dependencies represented with a spatial grid of latent parameters.

A. Data description

We used a portion of the Safety Pilot Model Deployment publicly accessible here : https://catalog.data.gov/dataset/safety-pilot-model-deployment-data.
All measures are taken with the Beam Steering Mechanism technology with a frequency of approximately 10 Hz (it is the largest frequency we could obtained from a public dataset). These measures include motion : speed and acceleration, and location : longitude and latitude. We took trajectories only passing in a defined perimeter centered in latitude 42.282970 and longitude -83.735390, all positions within latitude 42.282970 ± 0.003000 and longitude -83.735390 ± 0.003000 are kept.

It means that we track all positions inside a rectangle of approximately 600 meters vertically and 500 meters horizontally. You can see the corresponding portion of raod in Fig 1. It is the location with the largest number of registered positions, we kept only this trajectories because we want to learn spatial dependencies thus we need a high number of trajectories passing through the same locations. Also we keep only trajectories with at least 100 data points so that we have

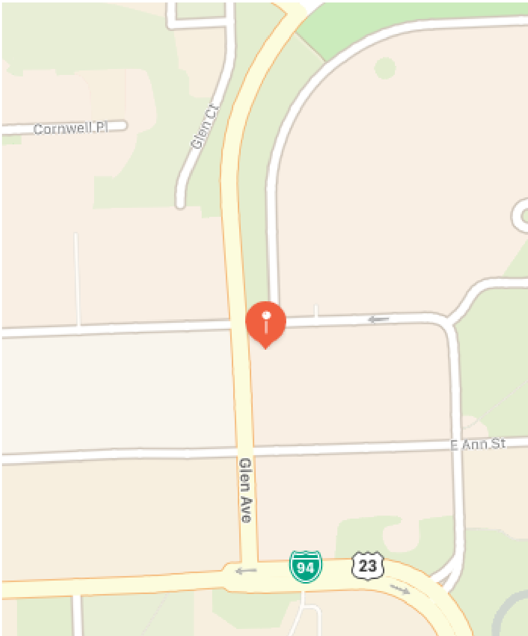

Fig. 1. Portion of road where belong sub dataset trajectories

enough dynamics to learn something.
This sub dataset has 266725 gps positions, it means approximately 7 hours of non-stop driving from approximately 300 drivers.

B. Model Settings and Baseline

Bl1 : first baseline, equation of motion of order 1.
Bl2 : second baseline, equation of motion of order 2.
Bl3 : third baseline, interpolation of input and last gps groundtruth position.
MR1 : motionresnet.
MRSRep1 : motionresnet and spatial representation parameters.
MRBi1 : motionresnet using Bi-RNN.
MRBiSRep1 : motionresnet using Bi-RNN and spatial representation parameters.

C. Results

We measure the reconstruction with the ground breaking truth between all the models : MSE score. From this we can derive RMSE and SMOOTH L_1 losses. We also look at three cases : straight line, stopped cars, complexed trajectories, marked as cat1, cat2 and cat3. You can see results in Table I.

We do the same experiments by varying the size of the mask : 10 means we only keep every tenth positions as inputs, 100 means we only keep every hundredth positions, etc... So we try to infer between 1s and 100s between each input position. You can see results in Table II.

Finally we also look at how the model spatial representation mesh width impacts model performances. You can see results in Table III.
Qualitative metric ? Show cherry picked inferred trajectories vs baseline ?

[H]

TABLE I

GROUND TRUTH IMPUTATION OVER THREE CASES (IN %)

	Bl1	**Bl2**	**Bl3**	**MR1**	**MRRep1**	**MRBi1**	**MRBiSRep1**
L_1MAE cat1	0	0	0	0	0	0	0
RMSE cat1	0	0	0	0	0	0	0
SMOOTH L_1 cat1	0	0	0	0	0	0	0
L_1MAE cat2	0	0	0	0	0	0	0
RMSE cat2	0	0	0	0	0	0	0
SMOOTH L_1 cat2	0	0	0	0	0	0	0
L_1MAE cat3	0	0	0	0	0	0	0
RMSE cat3	0	0	0	0	0	0	0
SMOOTH L_1 cat3	0	0	0	0	0	0	0
L_1MAE all	0	0	0	0	0	0	0
RMSE all	0	0	0	0	0	0	0
SMOOTH L_1 all	0	0	0	0	0	0	0
Avg	0	0	0	0	0	0	0

[H]

TABLE II

GROUND TRUTH IMPUTATION WITH VARIOUS MASK

	Bl1	**Bl2**	**Bl3**	**MR1**	**MRRep1**	**MRBi1**	**MRBiSRep1**
L_1MAE with mask 10	0	0	0	0	0	0	0
L_1MAE with mask 20	0	0	0	0	0	0	0
L_1MAE with mask 30	0	0	0	0	0	0	0
L_1MAE with mask 40	0	0	0	0	0	0	0
L_1MAE with mask 50	0	0	0	0	0	0	0
L_1MAE with mask 60	0	0	0	0	0	0	0
L_1MAE with mask 70	0	0	0	0	0	0	0
L_1MAE with mask 80	0	0	0	0	0	0	0
L_1MAE with mask 90	0	0	0	0	0	0	0
L_1MAE with mask 100	0	0	0	0	0	0	0
Avg	0	0	0	0	0	0	0

TABLE III
GROUND TRUTH IMPUTATION WITH VARIOUS SPATIAL REPRESENTATION MESH WIDTH (IN LATITUDE/LONGITUDE)

	MRRep1 5	**MRRep1 10**	**MRRep1 50**	**MRRep1 100**	**MRRep1 500**	**MRRep1 1000**
L_1MAE cat1	0	0	0	0	0	0
RMSE cat1	0	0	0	0	0	0
SMOOTH L_1 cat1	0	0	0	0	0	0
L_1MAE cat2	0	0	0	0	0	0
RMSE cat2	0	0	0	0	0	0
SMOOTH L_1 cat2	0	0	0	0	0	0
L_1MAE cat3	0	0	0	0	0	0
RMSE cat3	0	0	0	0	0	0
SMOOTH L_1 cat3	0	0	0	0	0	0
L_1MAE all	0	0	0	0	0	0
RMSE all	0	0	0	0	0	0
SMOOTH L_1 all	0	0	0	0	0	0
Avg	0	0	0	0	0	0

D. Future work

MSE is a bad metric for this kind of task we should find a way to modify the architecture so that the used loss is Cross-entropy.