Traj-ARIMA: A Spatial-Time Series Model for Network-Constrained Trajectory

Zhixian Yan
EPFL - Ecole Polytechnique Fédérale de Lausanne, Switzerland
zhixian.yan@epfl.ch

ABSTRACT

Trajectory data play an important role in analyzing real world applications that involve movement features, e.g. natural and social phenomena such as *bird migration*, *transportation management*, *urban planning* and *tourism analysis*. Such trajectory data are a speical kind of time series with another focus on the spatial dimension besides the temporal one. Traditional time series models, especially the ARIMA (*Auto-Regression Integrated Moving Average*) model, have provided sound theoretical backgrounds and promoted many successful applications for managing and forecasting time-relevant sequential data. This paper aims at extending the ARIMA model with spatial dimension, and further applying it for the network-constrained trajectory data. We implement and evaluate the model for trajectory database, in the context of traffic application scenario about vehicle movement constrained under a given network infrastructure. The proposed Traj-ARIMA model has many application perspectives, such as trajectory data regression and compression, outliers detection, traffic flow and vehicle speed prediction. In this paper, the major focus is on vehicle speed forecasting.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*time series database, spatial databases, GIS*; G.3 [**Probability and Statistics**]: Time Series Analysis—*Spatial-Time Series, Prediction*

General Terms

Algorithm, Experimentation, Verification

Keywords

Trajectory Databases, Time Series Models, ARIMA, Computational Transportation Science (CTS)

1. INTRODUCTION

With the advent of GPS and sensor-based tracking techniques, trajectory data become easily available and ubiquitous, both technically and economically. The recorded trajectory includes a sequence of data points, and can be considered as a typical scenario of time series application. Time series analysis comprises statistical methods (e.g. autocorrelation and spectral analysis) that attempt to understand sequential data and do forecasting [1] [7]. With sound theoretical backgrounds and sustaining software packages, time series have many successful applications in macroeconomics and finances. However, for trajectory data in moving object database, time series method is still a fancy and trial topic need to be further exposited. Even with a couple of studies focusing on similarity search [4], mining periodic patterns [8], detecting outliers [9] in time series databases, there are less interconnections with popular methods of time series analysis. In this paper, we study the conventional time series models, especially the time-domain driven analysis method ARIMA, and extend the model in the spatial dimension to analyze and predict network-constrained trajectories in the context of moving object database.

The rest of this paper is structured as follows: after a short introduction in Section 1, Section 2 reviews relevant conventional time series model, ARIMA in particular and its possible extensions for the spatial dimension, such as Vector-ARMA and ST-ARIMA; Section 3 addresses trajectories in a context of moving object database, and proposes the Traj-ARIMA model for network-constrained trajectories; the initial experimental results about vehicle speed analysis and prediction are presented in Section 4; and finally Section 5 points to conclusion and future work.

2. TIME SERIES MODEL AND SPATIAL EXTENSIONS

In this section, we briefly study and review ARIMA (Autoregressive Integrated Moving Average) model for time series, with the possible spatial extensions, such as the two major ones Vector ARMA and Space-Time ARIMA.

2.1 ARIMA Model

As shown by Box and Jenkins [1], time series can be considered as stochastic processes from the statistical point of view. Time series analysis provides models to represent the processes, in terms of many forms for modeling variations in the level of a process. Among those models, ARIMA (Autoregressive Integrated Moving Average) is a top-choice linear method. ARIMA combines the idea of the autoregressive (AR) model, the moving average (MA) model, and the integrated (I) model. Autoregressive process is regression on themselves, which means current value x_t is a linear combination of p (p is the order of AR) historical observations, plus a white noise ϵ_t; a moving average process of order q is a linear combination of current noise and q historical noises; ARMA combines them, referring to the model with p autoregressive terms and q moving average terms. The formulas of these three models are follows,

IWCTS'10, Nov 2, 2010 San Jose, CA, U.S.A.

$$AR(p):\ x_t = \phi_1 x_{t-1} + \ldots + \phi_p x_{t-p} + \epsilon_t = \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t \quad (1)$$

$$MA(q):\ x_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} = \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \quad (2)$$

$$\begin{aligned} ARMA(p,q):\ x_t &= \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} \\ &\quad + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \\ &= \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} \end{aligned} \quad (3)$$

By using the backshift operation B (i.e. $B(x_t) = x_{t-1}$), we can rewrite the AR(p), MA(q), and ARMA(p,q) models in a more compact way (see Formula 4-6).

$$\begin{aligned} AR(p):\ x_t(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p) \\ = x_t(1 - \sum_{i=1}^{p} \phi_i B^i) = \epsilon_t \end{aligned} \quad (4)$$

$$\begin{aligned} MA(q):\ x_t = \epsilon_t(1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q) \\ = \epsilon_t(1 + \sum_{j=1}^{q} \theta_j B^j) \end{aligned} \quad (5)$$

$$ARMA(p,q):\ (1 - \sum_{i=1}^{p} \phi_i B^i)\, x_t = \epsilon_t(1 + \sum_{j=1}^{q} \theta_j B^j) \quad (6)$$

ARMA can be further extended to ARIMA, with the combination of the integrated model (I) using the differencing operation. Before applying autoregressive and moving average, it takes d-level differencing. The differencing operation can transform non-stationary time series into a stationary one, which is very useful in analyzing the real-life time series data.

$$ARIMA: (1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d\, x_t = \epsilon_t(1 + \sum_{j=1}^{q} \theta_j B^j) \quad (7)$$

2.2 Spatial Extensions of ARIMA

The ARIMA model reviewed in Section 2.1 only discusses the temporal correlations among different observations in time series, without any consideration of the spatial correlations. For trajectory data in moving object database, however, spatial is another important issue, which cannot be overlooked in real world applications. For example, the vehicle speed in trajectory is not only related to the historical speed, but also affected by the facility (e.g. traffic flow) of the neighboring road network.

There are two major methods referring to spatial time series modeling, namely Vector ARMA and Space-Time ARIMA [10] [6]. The previous ARIMA model considers the temporal correlation, focusing on univariate time series. For Vector ARMA, it estimates the dynamic interactions among multiple time series, which can be considered as a subclass of the state-space model [10] [1]. The major difference is changing previously mentioned univariate into multivariate (vector), as shown in the following formula,

$$\begin{aligned} X_t &= \Phi_1 X_{t-1} + \Phi_2 X_{t-2} + \ldots + \Phi_p X_{t-p} + \varepsilon_t \\ &\quad + \Theta_1 \varepsilon_{t-1} + \Theta_2 \varepsilon_{t-2} + \ldots + \Theta_q \varepsilon_{t-q} \\ &= \sum_{i=1}^{p} \Phi_i X_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \Theta_j \varepsilon_{t-j} \end{aligned} \quad (8)$$

in which, $X_t = (x_{t1}, x_{t2}, \ldots, x_{tk})'$, $\varepsilon = (\epsilon_{t1}, \epsilon_{t2}, \ldots, \epsilon_{tk})'$ are vectors respectively for variants and white noises; and Φ_p and Θ_q are coefficient matrices need to be estimated in the experiment.

Space-time ARIMA (ST-ARIMA) can be approximately viewed as a special case of vector ARIMA, which emphasizes the spatial dimensions in terms of "spatial correlations", not only "temporal correlations". Concretely speaking, ST-ARIMA expresses each observation at time t and location l as a linearly weighted combination of previous observations and innovations lagged both in space and time [10]. Given a observation $X_t = (x_{t1}, x_{t2}, \ldots, x_{tl})'$, which means l observed-values at time t at l different locations, $W = [w_{ij}]_{l*l}$ is a $l \times l$ weighted matrix for l relevant locations. We get the following ST-ARMA model,

$$X_t = \sum_{i=1}^{p} \sum_{k=1}^{l} \Phi_i W_k X_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \sum_{k=1}^{l} \Theta_j W_k \varepsilon_{t-j} \quad (9)$$

which can be rewritten by applying backshift operator B,

$$(I - \sum_{i=1}^{p} \sum_{k=1}^{l} \Phi_i W_k B^i) X_t = \varepsilon_t (I + \sum_{j=1}^{q} \sum_{k=1}^{l} \Theta_j W_k B^j) \quad (10)$$

then we can further apply differencing operators to combine Formula (9) with the I (integrated) model and obtain the ST-ARIMA model as follows,

$$(I - \sum_{i=1}^{p} \sum_{k=1}^{l} \Phi_i W_k B^i)(1 - B^d) X_t = \varepsilon_t (I + \sum_{j=1}^{q} \sum_{k=1}^{l} \Theta_j W_k B^j) \quad (11)$$

3. TRAJECTORY ARIMA

In this section, we firstly discuss trajectory data management in the context of moving object database, especially with network constraints; afterward, we reconsider the previous ARIMA time series model and its spatial extensions, and adapt the models to the network-constrained trajectory data.

3.1 Trajectory Database

Trajectory data are usually detected by mobile or sensor devices, recording the position where a moving object temporally resides [12][13]. It can be formally defined as a sequence of spatiotemporal points $\langle space, time \rangle$.

DEFINITION 1 (TRAJECTORY). *A trajectory $\mathcal{T}$ is a sequence of spatiotemporal points $\langle space_i, time_i \rangle$ of a given object. In a conventional two dimension spatial system (not much difference with high dimensions), we get $\mathcal{T} = \{\langle x_i, y_i, t_i \rangle\}$ (with all i distinct and ordered, x_i and y_i are usually latitude and longitude in the context of GPS tracking data).*

Trajectories of a moving object often have route constraints in real world applications, which means vehicles can only move according to a certain network infrastructure. Not only do land vehicles like cars and buses have their certain route limitations; but also ships and airplanes have restricted trajectory paths. To analyze

trajectory data and do reasonable prediction, a comprehensible solution ought to consider the underlying network structure. Those network constraints for moving object trajectories can be defined as a network or a graph as follows,

DEFINITION 2 (NETWORK CONSTRAINTS). *A network constrain for trajectories is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ is the set of vertex $\{v_1, v_2, \ldots, v_n\}$, and $\mathcal{E}$ is a edge set for connecting vertex $\{e_1, e_2, \ldots, e_m\}$. $\mathcal{E}$ can be represented as a $n \times n$ matrix of vertex, i.e. $[e_{ij}]_{n \times n}$, in which e_{ij} can be $0, 1, \infty$ respectively for direct-connection, self-connection and no-connection .*

For a traffic road network, a road segment can be modeled as a node, and the connections among those road segments are edges. Figure 1 shows an example road network, and the following matrix is about the edge matrix connecting the road segments, where 0 means self-connection and ∞ means no connection. We can use the matrix for computing spatial lags from the spatial-time series viewpoint, which will be further discussed in Section 3.3.

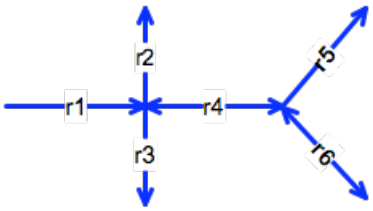

Figure 1: A example road network

$$M = \begin{array}{c} \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{array} \begin{array}{c} \begin{array}{cccccc} r1 & r2 & r3 & r4 & r5 & r6 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 1 & 0 & 1 & 1 \\ \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Real world trajectories should be consistent with the underlying network infrastructure, as an example shown in Fig. 2 (generated by [3]), which projects the trajectories of many moving object into the consistent network. Therefore, trajectories from Definition 1 need to be refined with network constraints. The original GPS tracked location (x_i, y_i) in a trajectory $\mathcal{T}$ need to be map-matched into road segments in a certain road network. With the underlying network, trajectory data analysis can consider the spatial correlations between the neighboring road segments.

DEFINITION 3 (NETWORK CONSTRAINED TRAJECTORY). *A trajectory under network constraints can be defined as $\mathcal{TN} = \{\mathcal{TS}, \mathcal{G}\}$, where $\mathcal{TS}$ is a set of trajectories, which might belong to one moving object or many different moving objects $\mathcal{TS} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m\}$; $\mathcal{G}$ is the constrained network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.*

For each trajectory $\mathcal{T}_k$, it is a sequence of GPS tracking data $\mathcal{T}_k = \{\langle x_{ki}, y_{ki}, t_{ki} \rangle\}$, as shown in Definition 1. By integrating network constraints and moving object information, the trajectory can be refined as $\mathcal{T}_k = \{\langle x_{ki}, y_{ki}, t_{ki}, mo\_id, road\_id \rangle\}$, where mo_id means ID of the moving object, $road\_id$ means the road network segments that the spatial location $\langle x_{ki}, y_{ki} \rangle$ can be matched.

3.2 Time Series for Trajectories

Before proposing a fully supporting spatial-time series model for trajectory database, we firstly just focus on the temporal correlations of sequential trajectory data. The main task is to transform the

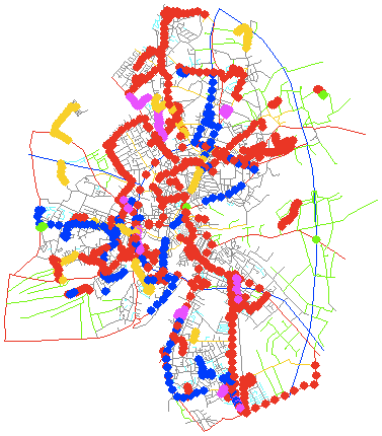

Figure 2: Network constrained trajectories

initial GPS tracking trajectory data $\langle x_t, y_t, t_i \rangle$ into a concrete time series. For trajectory data about moving object, one of the major sequential issues is about velocity analysis. With the availability of speed model for moving object, we can do further queries about speed forecasting and location prediction.

As GPS data are usually tracked very frequently in a short time interval (in our experiment case, tracking one record per second), we can approximately calculate the instant speed as the average speed between the previous node and the subsequent node as in Formula (12)[1],

$$s_i = \frac{\|\langle x_{i+1}, y_{i+1} \rangle - \langle x_{i-1}, y_{i-1} \rangle\|_2^2}{t_{i+1} - t_{i-1}} \tag{12}$$

Afterward, we can get the following trajectory speed time series (for each moving object):

$$\langle s_1, t_1 \rangle, \langle s_2, t_2 \rangle, \ldots, \langle s_n, t_n \rangle$$

Instead of analyzing the temporal correlations of trajectory instant speed at different time observations, we can also construct another time series model about distances, $\{\langle d_1, t_1 \rangle, \langle d_2, d_2 \rangle, \ldots, \langle d_n, t_n \rangle\}$ where $d_i = d_{i-1} + \|\langle x_i, y_i \rangle - \langle x_{i-1}, y_{i-1} \rangle\|_2^2$ and $d_0 = 0$. In this paper, we focus on using ARIMA and the extended spatial time series model for trajectory speed time series analysis. We can easily adapt univariate ARIMA model from Section 2.1 to construct the following ARIMA model:

$$(1 - \sum_{i=1}^{p} \phi_i B^i)\, s_t = \epsilon_t (1 + \sum_{j=1}^{q} \theta_j B^j) \tag{13}$$

We follow Box-Jenkins' typical steps to identify, estimate and diagnose an ARIMA Model for vehicle speed time series: plot the data and analyze the correlogram for variables; estimate parameters and fit the model; make diagnose checking and speed forecasting (mainly short-term forecasting). The experimental details will be presented later.

3.3 Spatial-Time Series for Trajectories

The previous ARIMA model for trajectory speed analysis only considers temporal dimension correlations, which means we model and forecast trajectory speed only based on its historical speeds,

[1]In some data sets, if the instant speed is captured by GPS devices, we can use is directly.

without using any knowledge about spatial neighborhood in the underlying network. However, in a real world application, spatial correlation is another important issue need to be considered for trajectory data analysis, especially in a context of network constrained trajectory data management. In other words, a correct vehicle speed prediction not only depends on the historical speeds, but also is influenced by the traffic flow status in the road segments nearby. Therefore, this section further investigates and extends Vector ARIMA and ST-ARIMA for trajectory data series.

For ST-ARIMA mentioned previously in Formula (11), all x_{ti} in a multivariate vector time series X_t belong to the same kind of time series with similar semantic meanings. However, the spatial correlations in trajectory speed time series cannot be modeled in the same way, because the spatial correlation between two trajectories is dynamic and affected by underlying network. We need to construct another speed time series about road segments nearby, so called *trajectory flow*. For example, to forecast a trajectory speed at $\langle x_i, y_i, t_i \rangle$, where location $\langle x_i, y_i \rangle$ can be determined at road segment r_5 in Figure 1, we get the following model,

$$s_t = F(\underbrace{s_{t-1}, s_{t-2}, s_{t-3}, \dots}_{historical\ speed(temporal)} , \underbrace{f_{r6}, f_{r4}, f_{r3}, \dots}_{trajectory\ flow(spatial)})$$

where, $s_{t-1}, s_{t-2}, \dots$ are historical speeds as temporal correlations, whilst $f_{r6}, f_{r4}, \dots$ are nearby road segment trajectory flows as spatial correlations. Hereinafter, we need to construct the time series for trajectory flow,

DEFINITION 4 (TRAJECTORY FLOW). *A trajectory flow is a time series belonging to a road segment, which records the average trajectory speed passing through this road segment. For each road segment, we get the flow time series $\mathcal{F} = \{\langle t_i, f_i \rangle\}$ (with all i distinct and ordered, f_i is the road capacity, in our experiment we use average speed).*

In stead of *Trajectory Flow* with the focus on the average passing speed at a road segment, we can also create a logically equal time series, by using *Traffic Flow* which considers how many vehicles passing through a road segment during a given time interval. For consistent, this paper applies *Trajectory Flow* time series.

In Section 3.1, network constraints are defined as a graph (road network), represented by a connecting edge matrix. We can determine the spatial lag matrix based on the connecting matrix.

$$M_{lag1} = M' = \begin{array}{c} \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{array} \begin{array}{c} \begin{array}{cccccc} r1 & r2 & r3 & r4 & r5 & r6 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & \infty & 1 & \infty & \infty \\ 1 & \infty & 0 & 1 & \infty & \infty \\ 1 & \infty & \infty & 0 & \infty & 1 \\ \infty & \infty & \infty & 1 & 0 & 1 \\ \infty & \infty & \infty & 1 & \infty & 0 \end{pmatrix} \end{array}$$

Then the weight for spatial lag 1 can be calculated with the equal weight for all the connecting road segments.

$$W_{lag1} = \begin{array}{c} \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{array} \begin{array}{c} \begin{array}{cccccc} r1 & r2 & r3 & r4 & r5 & r6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

We can apply Dijkstra's shortest path algorithm, for computing weights with more than one spatial lags. For example, the two lags connecting matrix and weight matrix are following,

$$M_{lag2} = \begin{array}{c} \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{array} \begin{array}{c} \begin{array}{cccccc} r1 & r2 & r3 & r4 & r5 & r6 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & 1 \\ \infty & \infty & 0 & \infty & \infty & 1 \\ \infty & \infty & \infty & 0 & \infty & \infty \\ 1 & \infty & \infty & \infty & 0 & \infty \\ 1 & \infty & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}$$

$$W_{lag2} = \begin{array}{c} \\ r1 \\ r2 \\ r3 \\ r4 \\ r5 \\ r6 \end{array} \begin{array}{c} \begin{array}{cccccc} r1 & r2 & r3 & r4 & r5 & r6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

When only considering trajectory flow time series, we can get the flowing model,

$$(I - \sum_{i=1}^{p} \sum_{k=1}^{l} \Phi_i W_k B^i)(1 - B^d) F_t = \varepsilon_t (I + \sum_{j=1}^{q} \sum_{k=1}^{l} \Theta_j W_k B^j) \quad (14)$$

For trajectory speed, it is more than just a vector time series as we need to use trajectory flow time series for modeling and forecasting trajectory speed time series. Therefore, we need to combine formula (13) and (14), respectively for temporal correlations on historical speeds and spatial correlations on nearby trajectory flows.

There are three possible combinational solutions,

1) Process trajectory speed time series and trajectory flow speed series together: for the road network in Fig. 1, there are 6 segments which means 6 trajectory flow series. Therefore, we can construct the time series model similar as Formula (14), but a new vector X, including 6 trajectory flows and 1 trajectory speed.

$$(I - \sum_{i=1}^{p} \sum_{k=1}^{l} \Phi_i W_k B^i)(1 - B^d) X_t = \varepsilon_t (I + \sum_{j=1}^{q} \sum_{k=1}^{l} \Theta_j W_k B^j) \quad (15)$$

2) Separately construct trajectory flow time series in advance, and then linearly plug it into the trajectory speed time series model.

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B^d)(s_t + \sum Wf) = \epsilon_t (1 + \sum_{j=1}^{q} \theta_j B^j) \quad (16)$$

3) Further refine 2), and consider the dynamic spatial (lags) weights for trajectory, as different road segments are involved with the evolution of the trajectory.

4. EXPERIMENT

This section shows the first results from our experiment, including model identification, parameter estimation, and diagnosis checking. We consider both real world traffic data set and simulated data set. At current step, for real world data set, we validate trajectory time series model, especially for trajectory speed modeling and forecasting; for simulated data set, it is used for the verification of spatial-time series model of trajectories.

4.1 Scenario and Data Set

Analyzing vehicle movement data is an important issue in traffic application. We apply and verify the proposed Traj-ARIMA model in two different data sets about traffic movement in a constrained road network. The first is a huge real world data sets, about tracking car movement in a Brazilian city; the second is data set generated by a simulation tool Brinkhoff generator[2].

1) **Real World Traffic Data** This data set is GPS tracking about car movement in Rio de Janeiro in Brazil. The tracked data are in regular form, one record per second. It is a good candidate for constructing trajectory speed time series. For example, we have one car with 827,330 GPS records $\langle x, y, t\rangle$ during more than one year. We divide the whole recording list into 364 trajectories, many of which follow the same path at different time. For example, Figure 3 shows five time series of trajectory speed, following the same movement route in approximately 4000 continuous seconds.

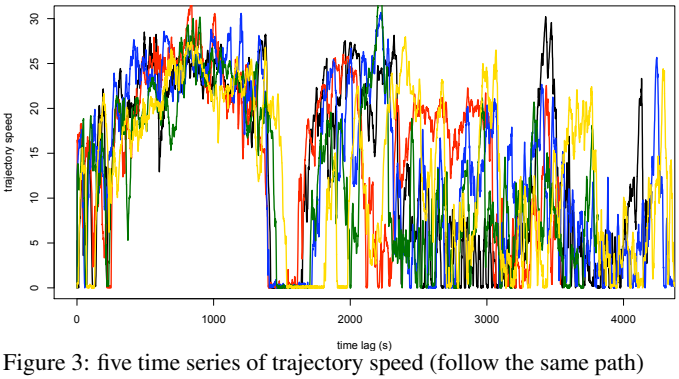

Figure 3: five time series of trajectory speed (follow the same path)

2) **Simulated Data set** As the previous data set does not have many cars moving at the same time, it is impossible to construct trajectory flow time series mentioned in Section 3.3. Furthermore, as real world initial GPS data are really dirty somehow, many researchers explore trajectory compression [5] and map-matching [2] techniques to clean the raw GPS data. Therefore, we plan to use some simulated traffic data for validating spatial-time series of trajectories. Brinkhoff generator is a popular opensource for generating spatiotemporal data under a given network constrain [3]. It combines real data (the network) with user-defined specifications of the properties (e.g. speed limitation, vehicle features) of the resulting trajectory dataset.

4.2 Time Series for Trajectories

The original Box-Jenkins ARIMA modeling procedure involves an interactive three-stage process, i.e. model selection, parameter estimation, and model checking [1]. For our case, we do two more explanations of the procedure, adding a stage of data preparation and a final stage of forecasting [11].

1) **Data Preparation** Data preparation includes transforming the raw GPS tracking data $\langle x, y, t\rangle$ into trajectory speed time series $\langle s, t\rangle$ by the formula (14). From the plot of the original trajectory speed time series and its autocorrelation function (ACF) at the upper of Figure 4, we can see it has long lags and need to be stationarized. Differencing operation is a key solution by introducing negative correlation. After one order of differencing, we get a new stationary time series shown at the bottom of Figure 4, together with a short ACF lag.

[2]http://www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/

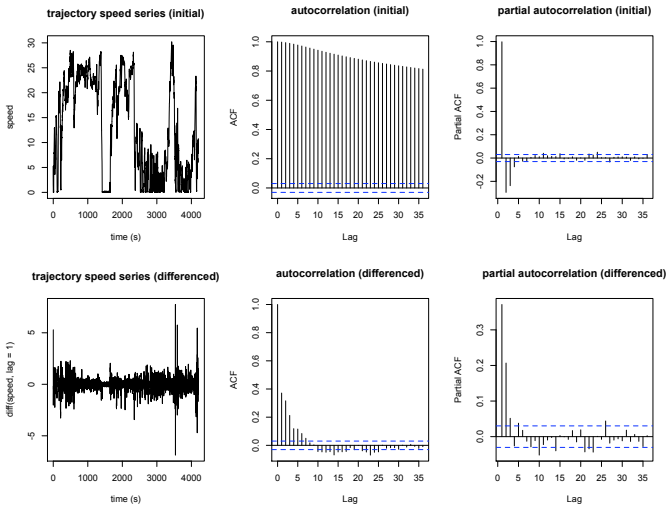

Figure 4: Speed time series ACF/PACF (original vs. differenced)

2) **Model Identification** After a time series has been stationarized by differencing, the next step is model selection which determines the order of AR (p) and MA (q) in fitting an ARIMA(p,d,q) model. From the partial autocorrelation (PACF) plot of the differenced series in Figure 4, we can see it "cuts off"at lag 2, which means it is significant at lag 2 and not significant at any higher order lags, therefore, we can tentatively identify the order of AR (p) is 2. From the differenced ACF plot, we identify the order of MA (q) is 6 as it "tails off"after lag 6. Therefore, a reasonable ARIMA model for the trajectory speed time series is ARIMA(2,1,6).

3) **Parameter Estimation** After determining the orders of ARIMA model, the next step aims at training the time series and finding the values of the model coefficients (i.e. ϕ_i and θ_j) which provide the best fit of the data. Two typical estimation methods are OLS (Ordinary Least Square) and MLE (Maximum Likelihood Estimation). Here, we apply MLE which is used a lot and usually has better estimation results in time series, by Formula 17,

$$\ell(\phi, \theta, \mu, \sigma^2; x_1, \ldots, x_n) = -\frac{1}{2}\{nlog\sigma^2 + log|V(\phi,\theta)| + \frac{(\mathbf{x} - \mu_{1\times n})V(\phi,\theta)^{-1}(\mathbf{x} - \mu_{1\times n})^T}{\sigma^2}\} \quad (17)$$

where $\{x_1, \ldots, x_n\}$ is the differenced trajectory speed time series, which is modeled as a linear function of white noise and has a joint Gaussian distribution $\mathcal{N}(\mu 1_n, \sigma^2 V(\phi,\theta))$; ϕ and θ are coefficients need to be estimated, together with μ and σ^2, by using the following optimization function,

$$\{\hat{\phi}, \hat{\theta}, \hat{\mu}, \hat{\sigma}\} = \underset{\phi,\theta,\mu,\sigma^2}{argmax}\{\ell(\phi, \theta, \mu, \sigma^2; x_1, \ldots, x_n)\} \quad (18)$$

By using R package for Statistical Computing[3], the estimated result for the ARIMA(2,1,6) model is as follows,

$$x_t = 1.5838x_{t-2} - 0.7359x_{t-1} + \epsilon_t - 1.2966\epsilon_{t-1} + 0.5590\epsilon_{t-2} - 0.0446\epsilon_{t-3} - 0.0078\epsilon_{t-4} + 0.1087\epsilon_{t-5} - 0.0115\epsilon_{t-6}$$

where the standard deviations of those parameters are respectively 0.0860, 0.0641, 0.0873, 0.0525, 0.0307, 0.0295, 0.0264, 0.0254; σ^2 is estimated as 0.3029 with log likelihood $= -3456.29$ and AIC = 6930.58.

[3]http://www.r-project.org/

4) **Diagnosis Checking** After specifying model and estimating its parameters, diagnose checking is concerned with testing the goodness of the model, whether it fits the real data set. Residual analysis is a typical method for model diagnostics, applying $\{residual = actual - predicted\}$. We compute and plot the diagnostic results in Figure 5, in which top-left is the standard residuals, we can see it looks like a typical normal distribution; top-right is the Q-Q (quantile-quantile) plot which is an effective tool for assessing normality; bottom-left is the ACF with clearly cut off at lag 1; and final bottom-right shows p-values are very close to 1. Those plots validate the good fitness of the model, but the Q-Q plot of the residuals shows not so perfectly well.

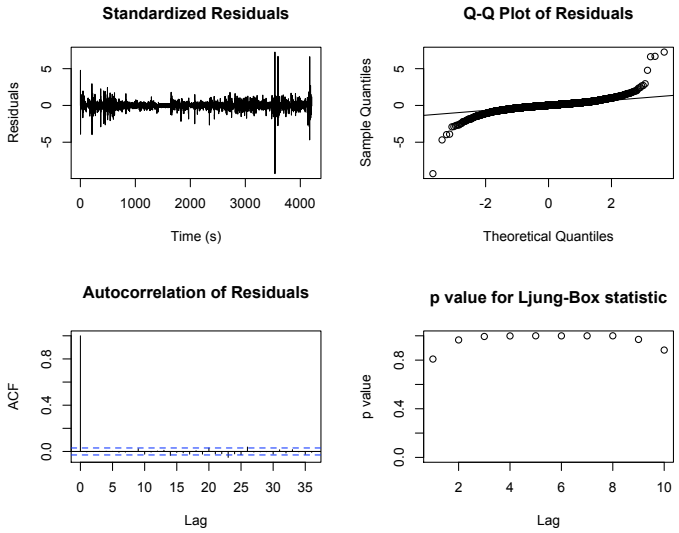

Figure 5: Diagnose checking plots of ARIMA

5) **Forecasting** One of the primary objectives of building a ARIMA model for time series is to forecast the values at future time. The following Figure 6 shows the forecasting results of the learned ARIMA(2,1,6) model. Have to say, the results are not so convincing. There are following possible reasons to explain this: (1) up to now, for this data set, it is still using one dimensional ARIMA model for trajectory data, which only focuses on temporal correlations, and there is no consideration about spatial correlations, that is why we need the spatial time series model for trajectory data; (2) building a ARIMA model for a whole trajectory is not so rational; my current research focus is on cutting trajectories into several semantic units "stops and moves", and then I apply the time series model for the separated move parts (a subsequence of a trajectory).

5. CONCLUSION

This paper has presented a spatial-time series model Traj-ARIMA for network constrained trajectory data, based on the extension of the conventional ARIMA model. To our knowledge, this is the first investigation on applying traditional time series methods for trajectory databases study. Besides a theoretical discussion on spatial time series modelling for trajectories, we validate the Traj-ARIMA model for the analysis of vehicle trajectories based on the typical time series experiment procedure. As vehicle velocity contains many uncertainty parameters in the real world systems, globally the prediction results we get from Traj-ARIMA are reasonable.

In addition to trajectory modelling and forecasting, we are able to discover semantic changes in the behavior of the vehicle trajectories, such as beginning a new trajectory or stopping for a while. In other words, when predicted results are far away from the real measures, there are two possible explanations: the presence of trajectory outliers and the change of vehicle behaviors. Therefore, our ongoing focus is on the application of Traj-ARIMA model for outlier detection, trajectory segmentation and stop identification, which are important issues for trajectory analysis.

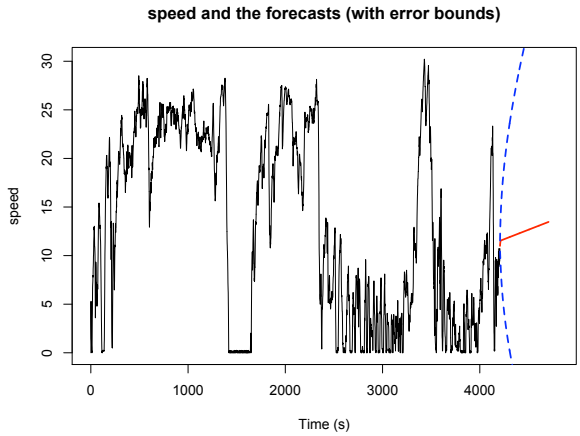

Figure 6: Trajectory Speed Forecasting

6. REFERENCES

[1] G. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 1994.

[2] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. In *VLDB*, pages 853–864, 2005.

[3] T. Brinkhoff. Generating Traffic Data. *IEEE Data Eng. Bull.*, 26(2):19–25, 2003.

[4] L. Chen. *Similarity search over time series and trajectory data*. PhD thesis, Waterloo, Ont., Canada, 2005.

[5] E. Frentzos and Y. Theodoridis. On the Effect of Trajectory Compression in Spatiotemporal Querying. In *ADBIS*, pages 217–233, 2007.

[6] R. Giacomini and C. W. Granger. Aggregation of Space-Time Processes. *Journal of Econometrics*, 118:7–26, 2004.

[7] J. G. D. Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443 – 473, 2006. Twenty five years of forecasting.

[8] J. Han, G. Dong, and Y. Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. pages 106–115, 1999.

[9] H. V. Jagadish, N. Koudas, and S. Muthukrishnan. Mining Deviants in a Time Series Database. In *VLDB*, pages 102–113, 1999.

[10] Y. Kamarianakis and P. Prastacos. Space-Time Modeling Of Traffic Flow. ERSA conference papers, European Regional Science Association, Aug. 2002.

[11] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman. *Forecasting: Methods and Applications*. WILEY, 1998.

[12] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. F. de Macêdo, F. Porto, and C. Vangenot. A Conceptual View on Trajectories. *Data Knowl. Eng.*, 65(1):126–146, 2008.

[13] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty. A Hybrid Model and Computing Platform for Spatio-Semantic Trajectories. In *ESWC*, pages 60–75, 2010.