STAT 5221 Final Project Report

Traffic Flow Forecasting During Covid-19 Pandemics

Prof Gabriel Young

Xinming Pan (xp2203)
Han Wang (hw2912)
Fangyang Zhang (fz2347)

Date: May 1, 2023

TABLE OF CONTENTS

Abstract

Our projects aims to develop a regional human flow prediction model with enhanced generalization capabilities and environmental adaptability using 90 weeks of data from New York City, which covers five features for 172 zip code areas(**regional population, median regional income, COVID-19 incidence rate, POI, and regional human flow**). To achieve this, we plan to use some typical time series model like AR, LSTM, HA etcs. to predict **the number of visits** in New York in future and try to apply a fancy temporal and spatial model called Attention Spatio-temporal Graph Convolutional Neural Network (ASTGCN) to effectively predict regional visitation volume. So, in this report, we are going to compare all these time series models with each other and make the prediction.

1. Introduction

With the acceleration of urbanization, predicting human traffic flow has become increasingly significant in urban planning, traffic management, and public safety. In this study, we utilized data from New York City and constructed a regional human traffic flow prediction model with enhanced generalization ability and environmental adaptability by applying classic time series models and a new model called ASTGCN.

2. Literature review

Urban area population mobility is a typical spatiotemporal data. When predicting the flow changes in a specific area for a future period, it is necessary to consider not only the historical flow change features of the area itself but also the spatial interaction information between different time periods with other related areas. That is, the correlation and heterogeneity between regions due to population flow.

In past transportation research, people mainly focused on predicting and analyzing the temporal dimension of regional visit volume, with the primary methods including ARIMA, AR, and LSTM, these are the models which are taught in class. However, in subsequent studies, people try to combine regional spatial dimension information in the predictions, including both fixed timestamp spatial information research and spatial dynamic information research under continuous time. Fixed timestamp spatial information research uses road, distance, regional similarity information for spatial adjacency matrix modeling, with the primary models including ST-ResNet, ST-3DNet under Euclidean distance, and Diffusion CRNN, ASTGCN under non-Euclidean distance. Spatial dynamic information research under continuous time is performed through adaptive learning of the original data to preserve the temporal and spatial causal relationships in the data, where Granger Causality Test is used more often. However, this method only considers the correlation of time series data and does not take other confounding factors into account. When considering regional heterogeneity, it is found that the spatial dependence relationship between regions is dynamically correlated. For example, the flow direction of entering and leaving the city during morning and evening peaks has significant differences. When predicting short-term visit volume, regional heterogeneity will have a more significant impact on the prediction. The purpose of this project is to compare the time series models we are taught in class with a fancy, advanced model, ASTGCN.

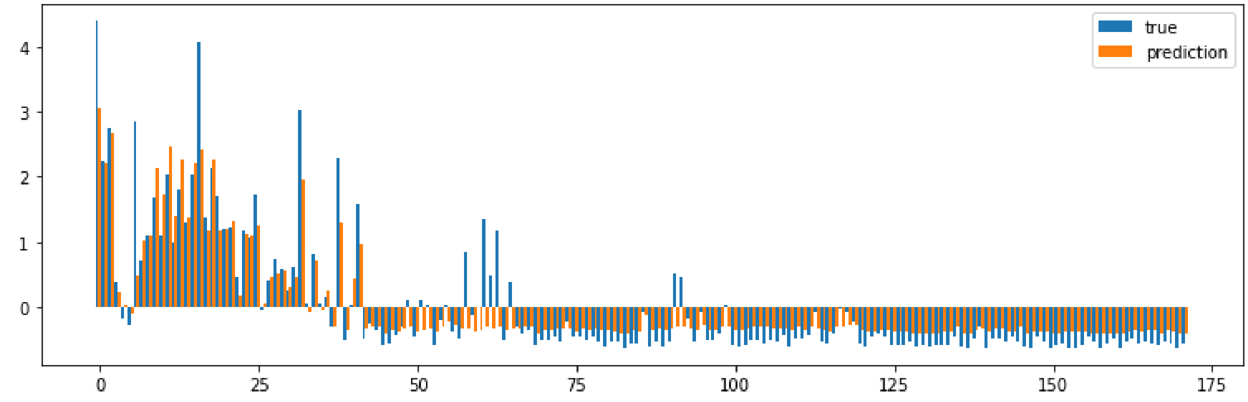
3. Experiment

3.1. Linear Regression

In linear regression regression, we are going to import our dataset first. After the data is loaded, we split the data into training and testing, then use the Sklearn package in Python and import linear regression to build multivariate linear regression and fit the model, because here we are going to use four features to predict **the number of visits** . The general formula for multivariate linear regression is as below:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m + \epsilon$$

Here, since we use four features to predict the **number of visits**, we will have 5 parameters in total. After we fit the model, then use the test data to get the accuracy of our model. Here is the figure that shows how the predicted data is fit to the actual data.

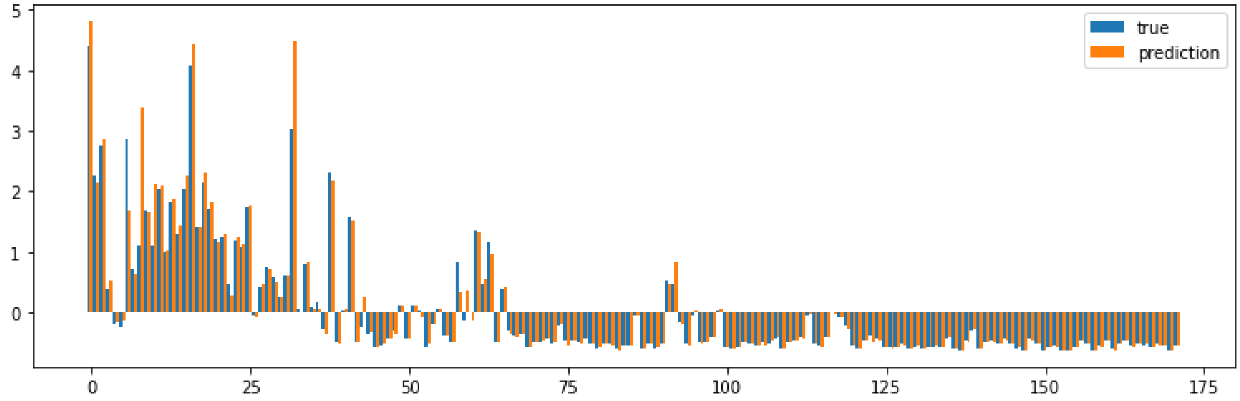


3.2. Autoregressive Model(p)

AR(p) model is the typical time series, so here we are going to AR(p) to make the prediction. Like the data loading before, we load the data first then split them into training and testing. Since AR(p) only needs the time feature. So, we only need **the number of visits**. The general formula for AR(p) is as below:

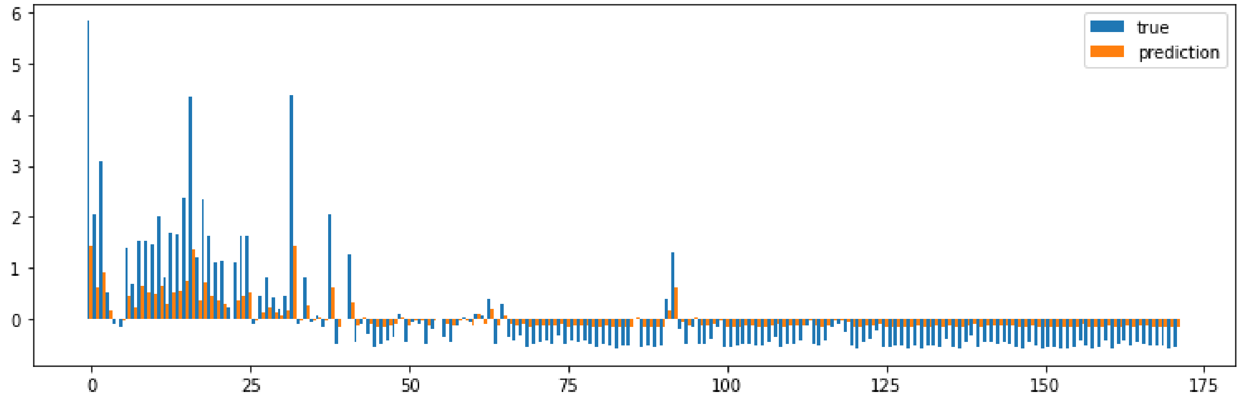
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t$$

The order ***p*** of the AR model is decided by ACF and ACVF figures, The autocorrelation function (ACF) is a measure of the correlation between a time series and a lagged version of itself. It shows how much a time series observation at a particular time point is related to observations at earlier time points. The ACF is calculated by taking the correlation between the time series and a lagged version of itself at different lag values. The autocovariance function (ACVF) is similar to the ACF, but it measures the covariance between a time series and a lagged version of itself, rather than the correlation. The ACVF shows how much the variance of the time series changes as the lag changes. Here, we use 90 weeks data to build the model and try the AIC test to find the most precise AR model to fix the data, and in addition, based on the PACF plot, we found that it is best in lags = 1, so we use AR(1) model in this data. Here is the figure that shows how the predicted data is fit to the actual data.



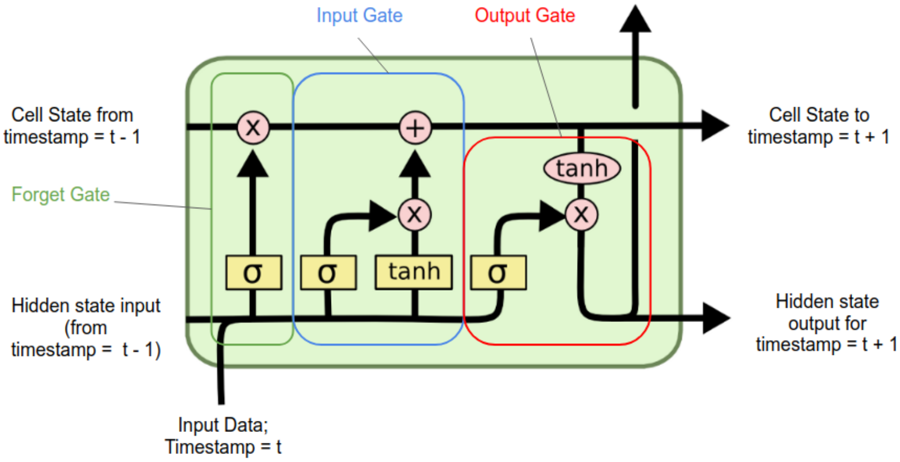
3.3. Historical Average

The HA model calculates the average of a time series over a specific period, such as the past few years, and uses that value to forecast future values. In our data set, we computed the number of visits by taking the average of visits in the past several weeks. Since it's one of the easiest models in time series analysis, the predicted result won't be very accurate. Here is the figure that shows how the predicted data is fit to the actual data.

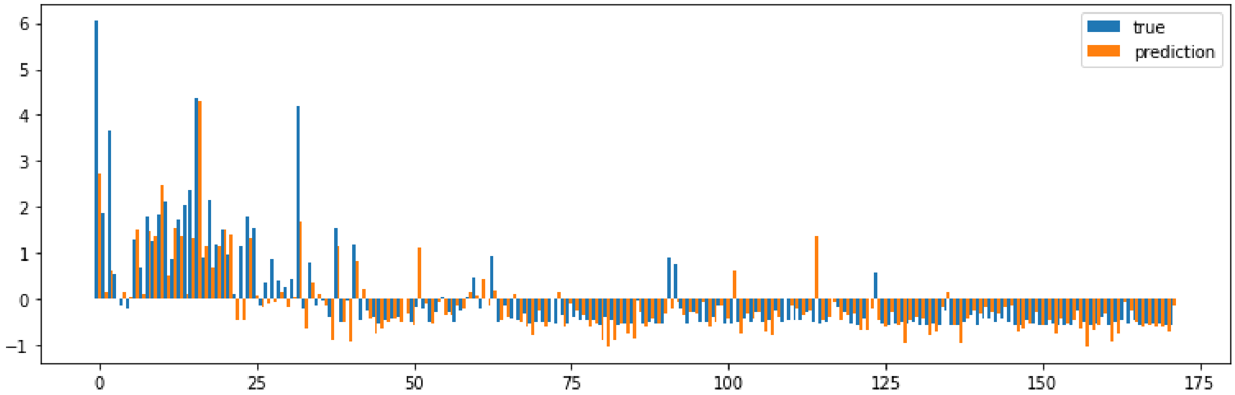


3.4. LSTM

LSTM stands for "**Long Short-Term Memory**", which is a type of Recurrent Neural Network (RNN) that is designed to handle the problem of the vanishing gradient in traditional RNNs. An LSTM network can be used for sequential data analysis and prediction, so here time series is typical sequential data. The three types of gates in an LSTM network are the input gate, the output gate, and the forget gate. The input gate determines how much new information should be added to the memory cells, while the forget gate determines how much old information should be retained. The output gate controls how much of the current state of the memory cells should be outputted to the next layer or to the final output. Here is the figure of one LSTM cell:


Input Gate
Output Gate
Cell State from timestamp = t - 1
Cell State to timestamp = t + 1
Forget Gate
tanh
σ
σ
tanh
σ
Hidden state input (from timestamp = t - 1)
Hidden state output for timestamp = t + 1
Input Data; Timestamp = t

LSTM networks are particularly useful for processing sequences of variable length and can capture long-term dependencies in the data. Here, we apply LSTM to the **number of visits** and make a prediction. Below is the figure that shows how the predicted data is fit to the actual data.



3.5. ASTGCN

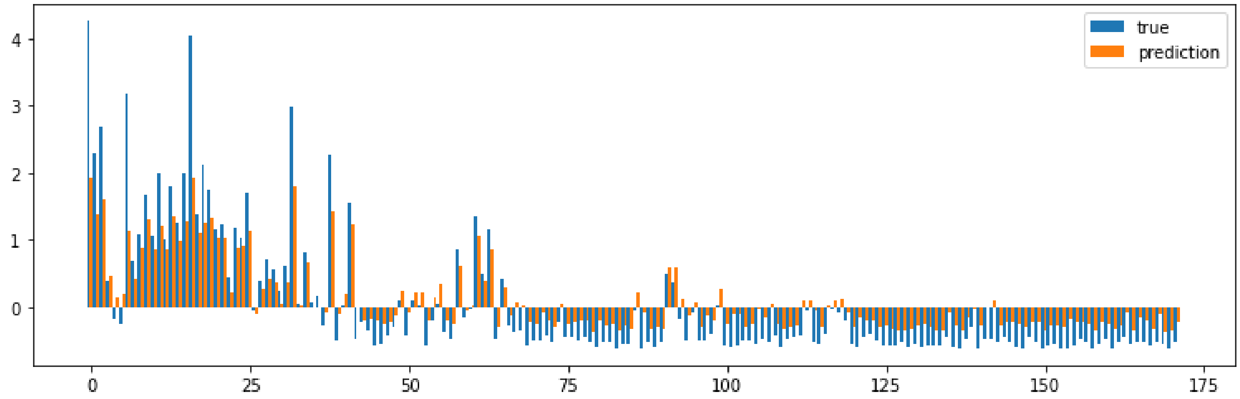
ASTGCN stands for "Attention Spatial-Temporal Graph Convolutional Network," which is a deep learning model designed for spatio-temporal prediction tasks. It combines graph convolutional neural networks (GCNs) and attention mechanisms to handle complex spatial and temporal dependencies in the data.

The ASTGCN model takes as input a spatio-temporal graph, which represents the relationships between different spatial locations and time points. The graph is represented as a set of nodes and edges, where each node corresponds to a spatial location and each edge represents the spatial or temporal relationship between two nodes.

The model consists of multiple layers of graph convolutions, which learn feature representations of the nodes in the graph. Each layer applies a set of filters to the graph, which capture different types of relationships between the nodes. The output of each layer is then fed into an attention mechanism, which weights the contributions of different nodes and edges based on their relevance to the prediction task.

The final output of the model is a prediction for each node in the graph at a future time point. The model is trained using a supervised learning approach, where the loss function is a measure of the difference between the predicted and actual values.

ASTGCN has been applied to a variety of spatio-temporal prediction tasks, such as traffic prediction, this is what we want to do in this project. It has shown to outperform other state-of-the-art methods in terms of prediction accuracy and efficiency. So, here we want to compare the performance of ASTGCN with the time series models we just built before. Below is the figure that shows how the predicted data is fit to the actual data.



4. Conclusion

Traditional time series model doesn’t take into account the spatial correlation between traffic data. This spatial-temporal correlation in traffic data prompts us to apply spatial temporal attention in a graph convolution model that may solve the prediction task. After we build each model, we are trying to compare their performance by calculating the MSE and Pearson Correlation of Coefficient. However, since Pearson R has drawbacks, then we are going to use MSE to get our best prediction model.

	HA	AR(p)	Linear Reg	LSTM	ASTGCN
MSE	0.515	0.202	0.311	0.394	0.197
Pearson R	0.951	0.901	0.834	0.792	0.963

Table 1). Performance of Each model

From the table above, we can see ASTGCN gets the best performance, because when applying the model to testing data, it gives us the lowest MSE. However, due to Occam's Razor Theorem, AR(p) is more simple than ASTGCN even if its performance is not as good as ASTGCN. So, finally, we conclude that AR(p) and ASTGCN can both predict the number of visits efficiently. Below are some findings and guessings.

- The reason why LSTM performs worse than AR(p) is maybe we need more training iterations. The optimizer will also affect the result.
- In Pearson R, two variables are not sampled independently from the distribution (bias in estimate), so we don't choose Pearson R to make the evaluation.
- All these models just consider the temporal correlation except ASTGCN.

5. Contribution

- **Xinming Pan**: Programming on ASTGCN model and making the evaluation of each model. In the report, work on the Conclusion and Evaluation part.
- **Han Wang**: Build the baseline models, make the EDA and do the data visualization. In the report, work on the Experiment part.
- **Fangyang Zhang**: Debug the code, make the PowerPoint and do the data cleaning. In the report, work on the Introduction and Abstract part.

6. Reference

- *Github Code*: https://github.com/panxinming/5221-Final-Project
- *AST-GCN: Attribute-Augmented Spatio temporal Graph Convolutional Network for Traffic Forecasting*, https://arxiv.org/abs/2011.11004
- Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks, https://arxiv.org/abs/1909.09586