NORTHEASTERN UNIVERSITY

# Markov-based Time Series Modeling for Temperature State of Traffic-Network

Rongshen Zhao, Esha Srivastava

December 7, 2021

**Abstract**: In this article, a Markov-based time series model is built to analyze the temperature of traffic-network. We carried out the steps, including finding a good data set, cleaning the data, choosing the states for temperature, calculating the empirical distribution, transition matrix, stationary distribution, auto-correlation function, comparing empirical and stationary distributions and checking the goodness of our model.

**Keywords**: Markov-based; time series; traffic-network; states; empirical distribution; transition matrix; stationary distribution; auto-correlation function.

## 1 DATA DESCRIPTION

The raw data(Figure 1.1) that we choose is "Metro Interstate Traffic Volume Data Set" from UCI Machine Learning Repository which is about traffic network conditions by integrating archived and real-time data under various external conditions, including holiday, temperature, cloud coverage and weather conditions between the year of 2012 and the year of 2018.

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | holiday | temp | rain_1h | snow_1h | clouds_al | weather_m | weather_d | date_time | traffic_volume | |
| 2 | None | 288.28 | 0 | 0 | 40 | Clouds | scattered | 10/2/2012 9:00 | 5545 | |
| 3 | None | 289.36 | 0 | 0 | 75 | Clouds | broken cl | 10/2/2012 10:00 | 4516 | |
| 4 | None | 289.58 | 0 | 0 | 90 | Clouds | overcast | 10/2/2012 11:00 | 4767 | |
| 5 | None | 290.13 | 0 | 0 | 90 | Clouds | overcast | 10/2/2012 12:00 | 5026 | |
| 6 | None | 291.14 | 0 | 0 | 75 | Clouds | broken cl | 10/2/2012 13:00 | 4918 | |
| 7 | None | 291.72 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 14:00 | 5181 | |
| 8 | None | 293.17 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 15:00 | 5584 | |
| 9 | None | 293.86 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 16:00 | 6015 | |
| 10 | None | 294.14 | 0 | 0 | 20 | Clouds | few cloud | 10/2/2012 17:00 | 5791 | |
| 11 | None | 293.1 | 0 | 0 | 20 | Clouds | few cloud | 10/2/2012 18:00 | 4770 | |
| 12 | None | 290.97 | 0 | 0 | 20 | Clouds | few cloud | 10/2/2012 19:00 | 3539 | |
| 13 | None | 289.38 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 20:00 | 2784 | |
| 14 | None | 288.61 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 21:00 | 2361 | |
| 15 | None | 287.16 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 22:00 | 1529 | |
| 16 | None | 285.45 | 0 | 0 | 1 | Clear | sky is cl | 10/2/2012 23:00 | 963 | |
| 17 | None | 284.63 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 0:00 | 506 | |
| 18 | None | 283.47 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 1:00 | 321 | |
| 19 | None | 281.18 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 2:00 | 273 | |
| 20 | None | 281.09 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 3:00 | 367 | |
| 21 | None | 279.53 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 4:00 | 814 | |
| 22 | None | 278.62 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 5:00 | 2718 | |
| 23 | None | 278.23 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 6:00 | 5673 | |
| 24 | None | 278.12 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 8:00 | 6511 | |
| 25 | None | 282.48 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 9:00 | 5471 | |
| 26 | None | 291.97 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 12:00 | 5097 | |
| 27 | None | 293.23 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 13:00 | 4887 | |
| 28 | None | 294.31 | 0 | 0 | 1 | Clear | sky is cl | 10/3/2012 14:00 | 5337 | |

Figure 1.1: Raw Data Set.

# 2 Data Preparation

First, we choose the column of temperature(Kelvin) to analyze meanwhile delete all the other columns and also select the data points from 2017-01-01 to 2017-12-31(Figure 2.1). Next, we visualize the data points of temperature by using Box Plot(Figure 2.2) and it is obvious that there exists outliers of our data points which are under 250 Kelvin. Then, we use IQR approach[1] to find the outliers and remove the corresponding rows. To be specific, we define the outlier value is above and below datasets normal range namely Upper and Lower bounds where the upper and the lower bound are:

$$\begin{aligned} Upper &= Q3 + 1.5 \times IQR \\ Lower &= Q1 - 1.5 \times IQR \end{aligned} \tag{2.1}$$

Then, we find two outliers which are in the row of 10565th and 10566th. After deleting these two outliers, we get the new data set(Figure 2.3).

| | temp | date_time |
|---|---|---|
| **0** | 269.75 | 2017-01-01 00:00:00 |
| **1** | 269.95 | 2017-01-01 01:00:00 |
| **2** | 269.75 | 2017-01-01 02:00:00 |
| **3** | 269.65 | 2017-01-01 03:00:00 |
| **4** | 269.48 | 2017-01-01 04:00:00 |
| **...** | ... | ... |
| **10576** | 250.46 | 2017-12-30 19:00:00 |
| **10577** | 250.68 | 2017-12-30 20:00:00 |
| **10578** | 251.23 | 2017-12-30 21:00:00 |
| **10579** | 251.15 | 2017-12-30 22:00:00 |
| **10580** | 251.38 | 2017-12-30 23:00:00 |

Figure 2.1: Data Points of Temperature in 2017.

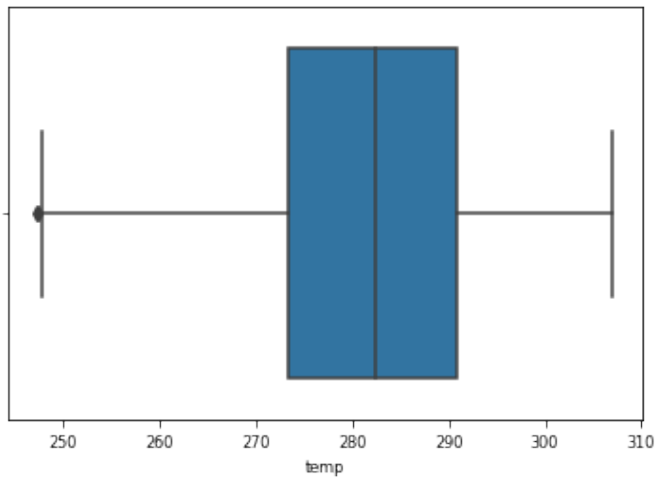


Figure 2.2: Boxplot-temp Column.

```
Old Shape:  (10581, 2)
New Shape:  (10579, 2)
```

Figure 2.3: Shape of Old and New Data Set.

## 3 Data Mapping

In order to map our data into a Markov chain, we first calculate the minimum and maximum and then get the range of our data points which is approximately 59.23. Therefore, we decide to choose 6 states and allocate each temperature to a specific state(Figure 3.1).

| | temp | states |
|---|---|---|
| **0** | 269.750 | 2.0 |
| **1** | 269.950 | 2.0 |
| **2** | 269.750 | 2.0 |
| **3** | 269.650 | 2.0 |
| **4** | 269.480 | 2.0 |
| **5** | 269.220 | 2.0 |
| **6** | 268.900 | 2.0 |
| **7** | 268.340 | 2.0 |
| **8** | 267.900 | 2.0 |
| **9** | 267.710 | 2.0 |
| **10** | 269.020 | 2.0 |

Figure 3.1: States Mapping.

## 4 Time Series Plot

For this step, we use Python to get the plot of time series of temperature(Figure 4.1) where the x-axis is the index of each data point and the y-axis is the value of temperature.

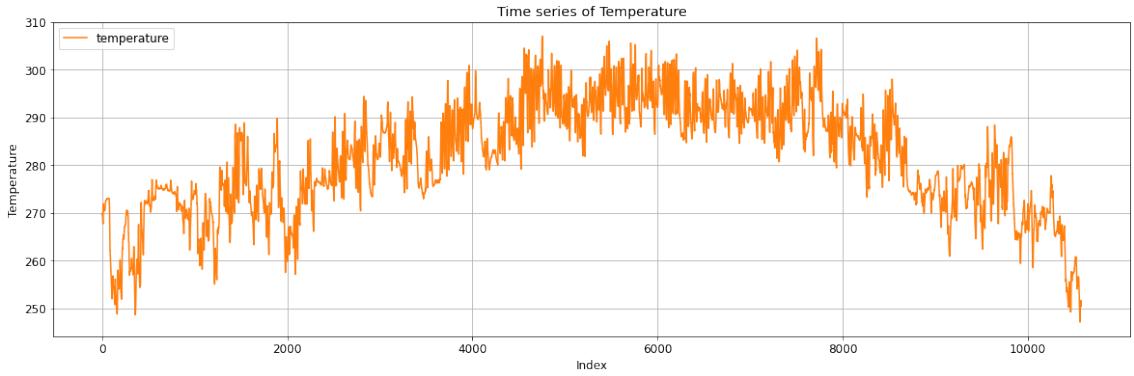


Figure 4.1: Time Series of Temperature.

## 5 Empirical Distribution Computation

Next, we calculate the number of data points corresponding to each state. And then we create a new dataframe including states and the number of these and also the relevant occupation frequencies which is the empirical distribution of our chain for each state(Figure 5.1). After this, we get the plot of fraction of time spent in each state where the x-axis is each state and the y-axis is the number of empirical distribution(Figure 5.2).

| | states | number | frequency |
|---|---|---|---|
| **states** | | | |
| **0.0** | 0 | 320 | 0.030249 |
| **1.0** | 1 | 924 | 0.087343 |
| **2.0** | 2 | 2972 | 0.280934 |
| **3.0** | 3 | 2655 | 0.250969 |
| **4.0** | 4 | 3066 | 0.289819 |
| **5.0** | 5 | 642 | 0.060686 |

Figure 5.1: Table of States.

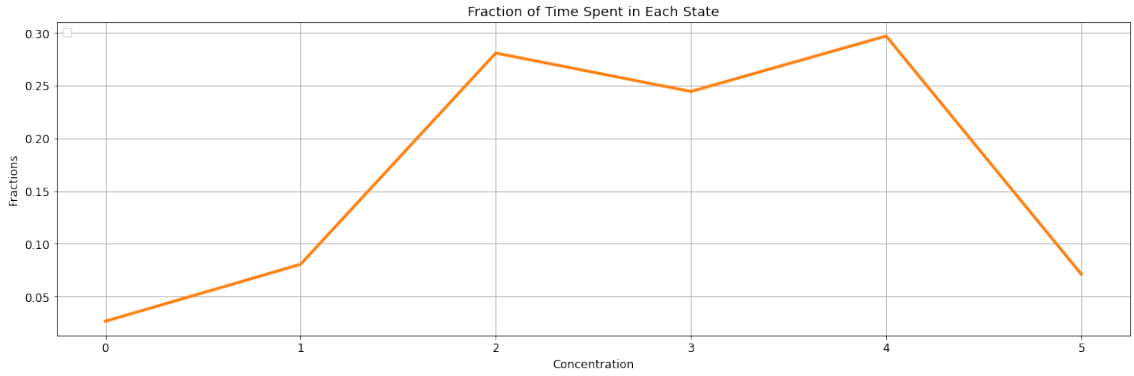


Figure 5.2: Frequency vs States Graph.

## 6 Transition Matrix Computation

To compute the transition matrix,which is essentially a representation of probability that our Markov state goes from one state to another, for this time series which follows the above distribution.We got the following transition matrix as a result

```
[[0.94984326 0.05015674 0.         0.         0.         0.        ]
 [0.01839827 0.93939394 0.04220779 0.         0.         0.        ]
 [0.         0.01345895 0.96433378 0.02220727 0.         0.        ]
 [0.         0.         0.02485876 0.93408663 0.04105461 0.        ]
 [0.         0.         0.         0.03555121 0.93868232 0.02576647]
 [0.         0.         0.         0.         0.12305296 0.87694704]]
```

Figure 6.1: Transition Matrix.

## 7 Stationary Distribution Computation

To compute the limiting probability vector a.k.a the stationary distribution which tell us the long term probability distribution of our markov states, we wrote a function which gave us the following result

```
array([0.03270157, 0.08914993, 0.27957765, 0.24975729, 0.28842028,
       0.06039329])
```

Figure 7.1: Stationary Distribution.

# 8 Comparison of Empirical Distribution and Stationary Distribution

In this figure given below we have shown the comparison between Empirical distribution from time series and the stationary distribution of chain:

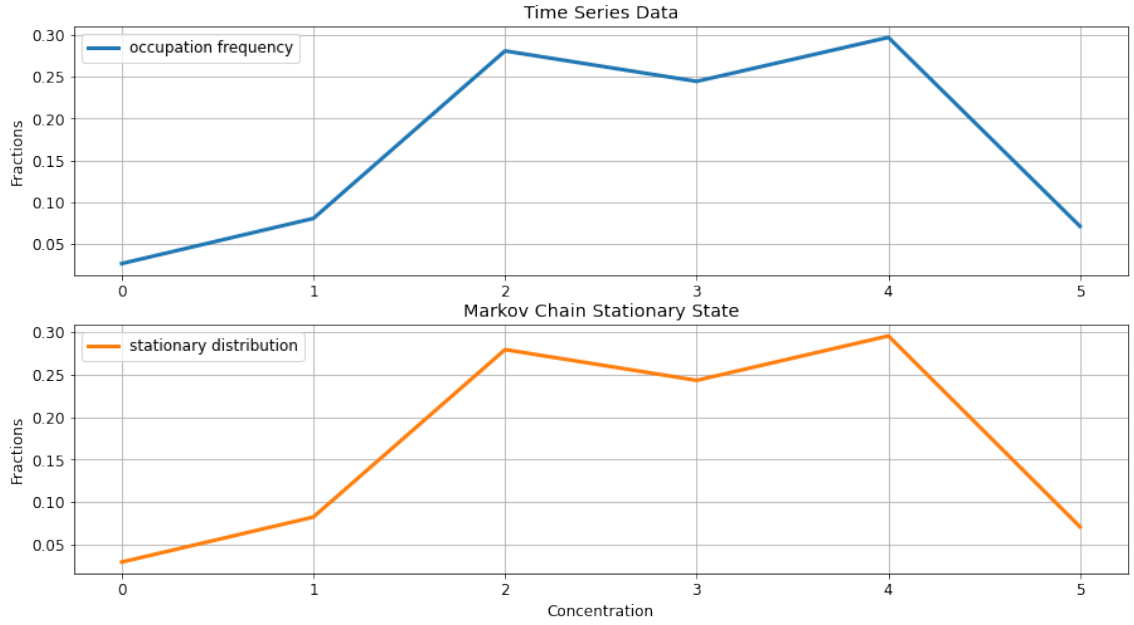


Figure 8.1: Occupation Frequencies vs Stationary Distribution.

The plot of the Markov State vs Normalized Frequency for original time series is similar which is an indication that the frequency distribution of the original data wasn't disturbed.

# 9 Simulation

For this step, we build a simulation of our Markov chain by using the transition matrix(Figure 6.1). And then we generate a typical time series and get the plot compared with the original time series(Figure 9.1). From these two plots, we could see that the plot of simulation is more stable than the original time series. Next, we need to evaluate our simulation by using auto-correlation function and chi test[2].

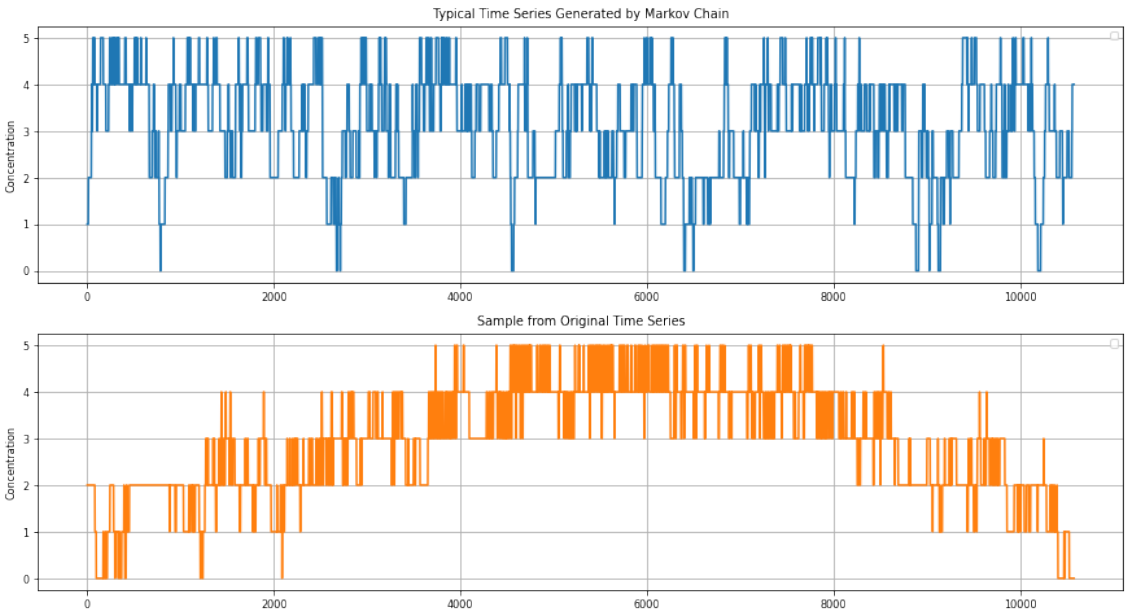


Figure 9.1: Simulation.

## 10 Auto-correlation Function

The correlation values for original time series and simulated time series are given in the following table: (we took K=30 , this is a sample of 6 values)

$$\begin{bmatrix} 1.0 & 1.0 \\ 0.978987377545787 & 0.981591011906189 \\ 0.95971090973076 & 0.963998218356827 \\ 0.941636395127476 & 0.947105020131268 \\ 0.924096081951632 & 0.930736518398575 \\ 0.907557396452242 & 0.915184211210319 \end{bmatrix}$$

Figure 10.1: Autocorrelation Matrix.

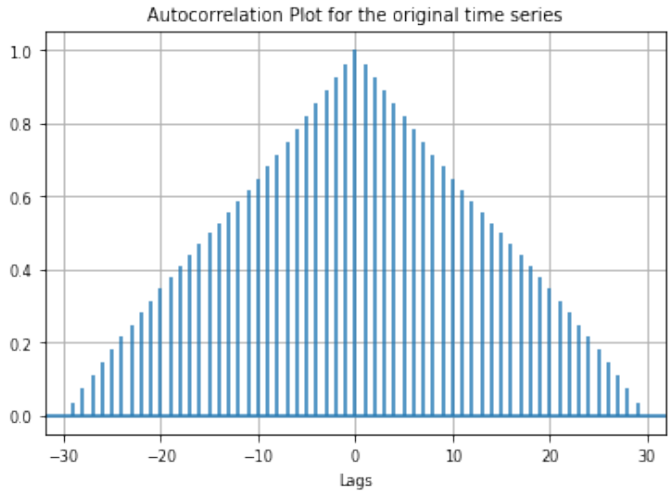


Figure 10.2: Autocorrelation Plot for the Original Time Series.

In these figures above we have made auto-correlation plots, Figure 10.2 and 10.3 represents the similarity between observations as a function of the time lag between them.

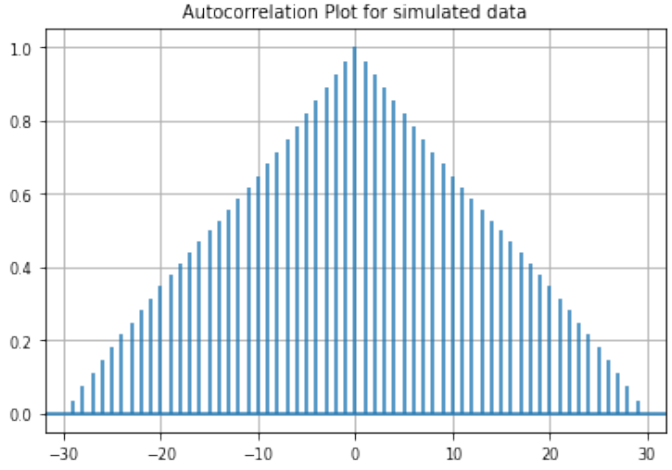


Figure 10.3: Auto-correlation Plot for Simulated Data.

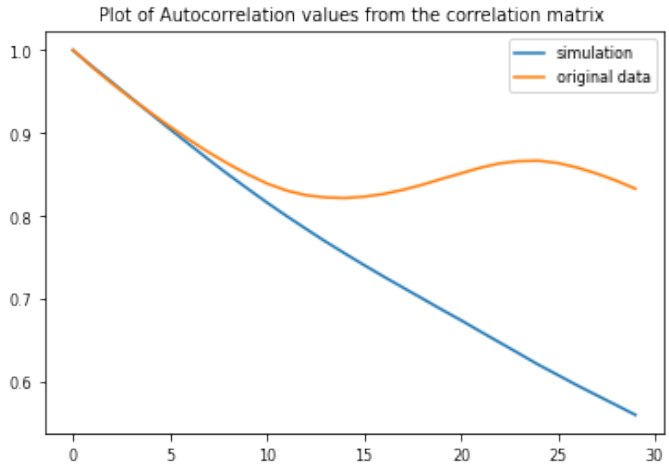


Figure 10.4: Plot of Auto-correlation Values from the Correlation Matrix.

Figure 10.4 plots the auto-correlation values from the correlation matrix of our original data and simulated data

## 11 Goodness of Fit

To check our "Goodness of fit" we ran chi test on our data set in this, we added together the rows of the matrix of expected frequencies and then we divided it by the number of states. This was our probability distribution,we then compared this with the same average of rows in observed frequencies.

```
[0. 014234875444839857,
 6. 067932897249324e-29,
 0. 0013477088948793193,
 0. 0,
 6. 583736171227538e-29,
 1. 7187097169070557e-29]
```

Figure 11.1: Result of Chi Test.

## 12 Conclusion

Since all the values from the chi test have p-value less than 0.05, we'll be accepting all the states. This implies that our dataset is a good fit to be a Markov chain .In a Markov process probability of each event depends only on the state attained in the previous event . For our dataset this makes perfect sense and is quite reasonable, as we have taken hourly data points of temperature , so if we know the current temperature we can easily make a prediction about the temperature in the next hour.

# REFERENCES

[1] Antoni Wilinski, "Time series modeling and forecasting based on a Markov chain with changing transition matrices," *Expert Systems With Applications 133 (2019) 163–172*, 2019. doi: https://doi.org/10.1016/j.eswa.2019.04.067

[2] Tao Li1,Jiaqi Ma2 and Changju Lee, "Markov-Based Time Series Modeling Framework for Traffic-Network State Prediction under Various External Conditions," *ASCE*, 2020.