Categorical Time Series Analysis:
Statistical models for time series analysis with categorical target variable

A research paper presented for the course

Prof. Dr. Münnich
M.Sc. Christopher Caratiola

Under the advisory of
Dr. Joscha Krause

Author Names

Gökhan Lüleci, 1613625
Anuj Patel, 1611010
Aarya Upadhye, 1607571
Pratibha Rajapaksha, 1622560

Data Science, M.Sc.
Universität Trier
Trier,Germany

20.03.2023

Contents

List of Figures

List of Tables

Categorical Time Series Analysis:
Statistical models for time series analysis with categorical target variable

Gökhan Lüleci, Anuj Patel, Aarya Upadhye, and Pratibha Rajapaksha

Trier University, Universitätsring 15, 54296 Trier, Germany

{s4goluel, s4anpate, s4aaupad, s4prraja}@uni-trier.de

Abstract

Contrary to numeric time series analysis, which has become rapidly popular nowadays, the field of categorical time series is still an emerging field. Despite the difficulties of the categorical data type, the proposed statistical models, visualization techniques, and statistical analyzes for categorical time series enrich the literature day by day. This paper, it is aimed to inspect various statistical analyses and methods for the Categorical Time series by comparing all aspects and interpreting the research findings. With the help of several motivation examples and real dataset implementation, the basic principles of statistical analysis techniques in categorical time series and statistical models will be evaluated comparatively. The assessment of the results, which were obtained after applying models to a real data set, shed a light on the theoretical comparison. In addition to all these, different ways of applying the statistical models and advanced further research areas will be also mentioned.

Keywords— Categorical Time Series Analysis, Spectral Envelope, Rate Evolution Graph, Markov-Chain Models, Hidden Markov Model, NDARMA Model, Multinomial Logit Model

1 Introduction

1.1 Introduction to Time Series Analysis

Time series analysis includes techniques for evaluating time series data to derive useful statistics and other properties at regular time intervals. It is a basic method for figuring out how metric changes over time and figuring out what it will be in the future. Time series methods are used by analysts in a wide range of situations. It is defined as a list of quantitative observations that are ordered by time. According to the traditional linear regression model, it was thought that the residuals of the equations being solved are randomly independent. Because

of this, methods that don't depend on time and can be used with both cross-sectional and experimental data were used.[10][17]

We get the forecasts based on unknown data and on the basis of the observations the future is anticipated. In a time series, the dataset changes. A time series shows that the observations are linked in a clear way. In a time series, you can have any variable that changes over time. It is commonly used to track how things are going over time. This can be kept an eye on for a short time or for a long time. People usually expect time series to be made at even intervals. When the data in a time series are both regular and timed, we call it a "regular time series." If the data are neither regular nor timed, we call it a "irregular time series."[17][16]

There are several application areas for time series analysis, including Economic Forecasting, Sales Forecasting, Budgetary Analysis, Stock Market Analysis, etc. These are a few examples of real-world applications - Prediction of the closing price of the stock, Prediction of the unemployment rate of a particular city.

1.2 History - Background of Time Series Analysis

In the early natural sciences, time series were already very important. Babylonian astronomy looked at the positions of the stars and planets to predict what would happen. Time series analysis helped find patterns in the observations of a variable and draw "rules" from them, or use all the information in this variable to predict how things will change in the future. These processes, which were also used by the Babylonians, are based on the idea that a time series can be broken down into a finite number of independent, non-observable parts that change over time and can be predicted in advance. For this technique to work, the variable must be affected by different, independent factors. Charles Babbage and William Stanley Jevons used this method to study astronomy around the middle of the 19th century. Warren M. Persons was the first person to break down a time series into unobserved causal components(1919).

Four different components were recognized as follows:

- a long-run development, **the trend**,
- **the business cycle**, with periods of more than one year,
- **the seasonal cycle**, that contains the ups and downs over a year, and
- a component that contains all movements which do not belong to any of the following - the trend, the business cycle, the seasonal component, **the residual**

These concepts were systematized and generalized by Herman Wold (1938). Box and Jenkins (1970) devised methodologies for the empirical application of these notions. They got rid of many parts and switched to a single stochastic model for making time series. Statistical data are used in this method to figure

out a certain model. The model's parameters are thought to be. Tests of statistics look at the details of the model. This step is repeated until a model meets the requirements. Last, this model can be used to make predictions.[15]

1.3 Main Concepts of Time Series Analysis

1.3.1 Stochastic Process

Probability theory is used to come up with theoretical approaches for time series. Let's say that the T-dimensional vector of random variables $x_1, x_2, ..., x_T$ is given with the relating multivariate distribution. It could also be called a bunch of random variables such as $[X_t]_{t=1}^{T}$ known as the stochastic process or data generating process (DGP). As per the concept, there could be any number of realizations of a process because they all come from the same data and have the same statistical properties.[15]

1.3.2 Ergodicity

Ergodicity is the idea that the sample moments can be calculated from a limited number of observations. They come together at the same time as the majority of the population for $T \to \infty$. This is true only if we can assume that the expectations $E[X] = \mu$ and the variances $V[X_t] = \sigma_x^2$ are constant for all 't'. Random variables must be considered to have these characteristics, which are termed consistency properties.[15]

1.3.3 Stationarity

For a stochastic process to be ergodic, it must be stationary, or in statistical equilibrium. The process is said to be strictly stationary if we assume that a change in time doesn't change the common distribution function of the random process. But it's hard to put this idea into practice in real life. Because of this, we only look at stationarity in the second moment, which is why it's called "weak stationarity." Here are the definitions of stationarity for every single moment of a stochastic process -$[X_t]$:[15]

1. **Mean Stationarity:** A process is mean stationary if $E[X_t] = \mu_t = \mu$ is constant for all 't'.

2. **Variance Stationarity:** A process is variance stationary if $V[X_t] = E[(X_t - \mu_t)^2] = \sigma_x^2 = \gamma(0)$ is constant and finite for all 't'.

3. **Covariance Stationarity:** A process is covariance stationary if $Cov[X_t, X_s] = E[(X_t - \mu_t)(X_t - \mu_s)] = \gamma(|s - t|)$ is only a function of how far apart in time two random variables are, and it has nothing to do with the time 't' itself.

4. **Weak Stationarity:** A stochastic process is weakly stationary when it is both mean and covariance stationary.

1.3.4 Random Walk

The stochastic process $[u_t]$ is said to be a pure random or a white noise process, only if it has the properties listed below:

$E[u_t] = 0$ and $V[u_t] = \sigma^2$ for every t, as well as $Cov[u_t, u_s] = E[u_t u_s] = 0$ for all $t \neq s$. Apparently, this process is weakly stationary.

- **The definition of stochastic process is** $[X_t]$

$X_t = \begin{cases} u_1, & \text{for } t = 1 \\ X_{t-1} + u_t, & \text{for } t = 2, 3, \ldots, \end{cases}$ where X_t is a pure random process. This stochastic process, a random walk without drift, can be rewritten as: $X_t = \sum_{j=1}^{t} u_j$

Let's say we come up with u_t by tossing a fair coin. We get heads 50 percent of the time (our random variable has the value +1 in this case) and tails 50 percent of the time (in this case, our random variable has the value -1). Let's look at a situation in which $X_0 = 0$ when $t = 0$. It's quite easy to observe that all possible realizations (time series) of this random walk can only take values within the area shown in Figure, which is limited by the two angle bisectors. If every time you flip a coin, you get heads (tails), the time series would be +1 (-1) for t = 1, +2 (-2) for t = 2, and so on.

We generate u_t by flipping a fair coin (heads = +1,tails = -1).

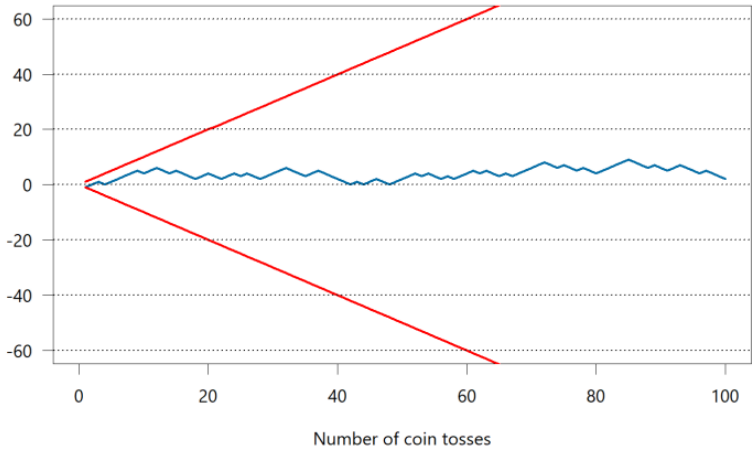


Figure 1: Random Walk

First two moments of a random walk:

$E[X_t] = E[\sum_{j-1}^{t} u_j] = \sum_{j-1}^{t} E[u_j] = 0$,

$V[X_t] = V[\sum_{j=1}^{t} u_j] = \sum_{j=1}^{t} V[u_j] = t\sigma^2$, and

$Cov[X_t, X_s] = E[(\sum_{j=1}^{t} u_j)(\sum_{i=1}^{s} u_i)] = \sum_{j=1}^{t} \sum_{i=1}^{s} E[u_j u_i] = min(t, s)\sigma^2$.

So, a random walk without drift is stable in terms of the mean, but not in terms of the variance or covariance. This means that it does not simply stand

there weakly. The random walk without drift is an important part of a group of stochastic processes that are not static and can be used to describe how economic time series change over time.[15]

1.4 Dealing with Categorical Data Type

1.4.1 Categorical random variable

A categorical random variable can be defined as a discrete random variable whose function range is a categorical set. Its extent from Binary random variable to multinomial random variable. For an example,

Dice roll experiment:

$X_{(DiceOutcome)}$= $\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}$

For the m category , the notations as follows:

Range of X is coded as V = $\{0,...,m\}$

$$P(x=0) = 1 - \sum_{j=1}^{m} P(x=j)$$

1.4.2 Quantitative and Qualitative Time Series

Time Series or time-stamped data, is a pattern in a consecutive data point indexed over time. Time series forecasting predicts future values based on previously observed values with a help of a model. It can be useful to observe the change of a given variable, asset, or security over time. It can also be used to examine how the changes associated with the chosen data point compared to the alterations in the values of other variables over the same time period.

There are different types of time series problems. Qualitative and Quantitative are the major 2 types of time series. The Qualitative Time Series can be defined as time series whose function range is a categorical set, whereas else Quantitative Time Series has a numerical set for function range. Qualitative time series can be categorized in ordinal and nominal time series that will be discussed in the upcoming topics in this.

1.5 Appropriate Data for Categorical Time Series Models and Selection

Following instances can be recognized as data for the categorical time series:

Genetic or protein sequences

A use case of a nominal categorical time series data depict here. DNA sequence data where there are four nucleotide and they are one letter acronym as follows, A is for adenine, G is for guanine, C is for cytosine, T is for thymine. Hence DNA types is represented as a sequence of letters from $\{A, C, G, T\}$ and

can transformed to normal categorical time series by allocating A=1, C=2, G=3 and T=4.

Statistical Process Control (SPC)

The SPC method is used mostly in production to reduce variability. To control this inconsistency, data are gathered from different applications and perform analysis. Data in SPC can be either ordinal or nominal. For example: hottest to coldest and lighter to heaviest we can encode in such a way that data is ordinal. At the same time, defects of a ceiling fan ('poor covering', 'bubbles', etc.) are an example of nominal data.

X_t = result of inspection of item

X_t = i for i = 1, . . . , m iff item has non-conformity type 'i'

X_t = 0 iff conforming

Selected dataset and utilised tools

The study utilized a specific dataset consisting of sequence data in FASTA format for a bacterial genome with the accession number CP009973.1., in parallel with researches in the literature. This dataset is curated and maintained by the European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL-EBI) and comprises 4 nucleotide bases - Thymine (T), Cytosine (C), Adenine (A), and Guanine (G). The organism under study is Yersinia pestis CO92, belonging to the enterobacteria group, with a total sequence length of 4,722,852. [9] The data can be accessed at ENA Webpage. https://www.ebi.ac.uk/ena/browser/api/fasta/CP009973.1

We employed various programming languages for different purposes such as modeling and visualization. To determine the stability and suitability of the data, Julia was utilized for Rate Evolution, which revealed that our dataset was compatible with the model. Python was employed for Markov and Hidden Markov modeling, as well as for generating the corresponding results. Since R was the programming language of choice for Multinomial Logit modeling in most reference papers, we also choose R for this task. Finally, for visualizing the spectral envelope, we used R.

1.6 The Confronted Challenges in Time Series Analysis

- All types of time series data, whether they are categorical, discrete, or numeric, face a common challenge. This challenge arises because the observations in a time-stamped dataset are typically continuous, occurring at regular intervals such as hourly or daily. Inconsistencies in the observation intervals can lead to misinterpretation of the results.

- When dealing with categorical time series data, one of the main challenges is that the presence of numerous experimental variables makes it highly improbable to discover a predictive model.
- In contrast to Quantitative time series, it is impossible to do arithmetic operations for categorical range for example; mean, variance, etc calculations. Furthermore, the data range is unordered hence it is not possible to calculate quantiles.

1.7 Introduction to Statistical Models

There arises a large amount of data from several practical fields which need to be handled. As explored in the past literature, several models have been applied to such real-time-series data over the last few decades. Out of these, this research is focused on a few important models namely, Hidden Markov Models, Multinomial Logit Models, NDARMA models, and Markov chains. Alternative methods of formulation of each of these models are discussed. In the next sections, we will study the formulation of these models, implementation of categorical time-series data, analysis of their performance, and brief comparison. We also choose such a dataset and apply our models to that dataset, to study their results.

Let us have a brief look at the history of the models. Firstly we review the history of one of the oldest and most popular models for discrete time-series data, i.e., NDARMA models. Since 1976, the theory of ARMA models (Box & Jenkins, 1976) has been playing a vital role in the modeling of continuous time series. Later in 1983, Jacobs & Lewis proposed the 'new' discrete ARMA models. Conventional ARMA models could be easily adapted for categorical data, by offering some kind of counterpart [23]. Over the last couple of decades, research has been continued in formulating discrete ARMA models in different ways. An alternative method of the formulation was demonstrated by Biswas & Song (2009). It extended the use of Pegram's operator to define discrete-valued ARMA processes, which was originally proposed only for AR processes, i.e. for modeling of continuous time series data. The proposed model is able to analyze any type of discrete data. It is applied to several real datasets, for instance, a small sample from Infant sleep data by Stoffer et al. (1988). Analyses showed that the performance of these models was slightly better for certain cases.

Markov chains have been widely used as models for stationary discrete time series. They provide great insights along with more information about predictions. One-step transition probabilities hold a major point of strength about these models [26]. However, in case Higher-order Markov chains are used, this leads to the problem of over-parametrization. Thus, these models are over parametrized for statistical purposes [13]. Further, the problem arises when the data to be modeled is non-Markovian, or not first-order Markovian.

Another important type of model is regression models, which are widely applied to time-series data. Regression theory for solving time series is based upon a simple theory of generalized linear models and partial likelihood inference. They have proved very useful for this particular approach [26]. These

models are very popular because of their many advantages over other methods. One of the key advantages is their ability to easily incorporate time-dependent covariate information [23]. They allow parsimonious modeling as opposed to other procedures. They do not rely on the Markov assumption and the notion of stationarity. It is evident that regression models allow both positive and negative associations to be taken into account by a suitable model parametrization, thus making it a versatile problem-solving approach. The behavior of these models on categorical data is flexible. Past literature has employed regression methodology on different applications and obtained very good results.

Given the past data

$$y_{t-1}^{*} = (y_{t-1}, y_{t-2}, ..., y_1)$$

and possible covariates

$$x_{t-1}^{*} = (x_{t-1}, x_{t-2}, ..., x_1),$$

the regression approach seeks to model the conditional distribution of Y_t.

A general regression model for categorical time series is also explained by Zhen (2008).

We conclude this section by pointing out that, in general, the behavior of different methods varies according to the nature of the data.

1.8 Visualisation and Analyses of Categorical Time Series

As already mentioned in the first chapter, mathematical operations, and plotting might be barely implemented by some tailor-made solutions for the ordinal data type. These constraints are even worse if the categories are nominal because of the lack of superior nature between categories. The necessary concepts such as distance measurement, autocorrelation, mean or variance are very demanding for the nominal categories. Since this paper uses the Nominal Dataset -DNA Sequence- for statistical models application, only the visualization techniques and analysis related to nominal categories will be mentioned.

The first and powerful method is assigning numbers to nominal categories, to calculate the above-mentioned concepts and the relations between categories. But unfortunately, this compels to have ordinal relationships between categories, which in practice does not exist.[19] To explain this concept, we have extended the Ribler's example. Although the main dataset of this paper is DNA Sequence from European Nucleotide Archive, our motivation example for visualization was designed by us for presentation purposes. After the motivation example, in the following chapters, implementation of visualization techniques will be implemented on DNA Sequence.

Our motivation example will be the CMYK Color Model. This subtractive color model consists of four main color types. (Cyan, Magenta, Yellow,

Key/Black) This model might be seen as the opposite of additive color models such as RGB. That means white is the base/original color of the printable ground and the black is derived from a full combination of other colors. The names of colors have been considered as a categorical data type. Thus, it is appropriate for Categorical Time Series analysis. To clarify the demanding sides of Categorical Time Series Analysis by visualization, the colors are mapped in four different ways.

Color Name	Natural Mapping	Random Mapping	Shades of..	Mapping by Shade	Key-Black or Not
Pine green	1	1	Cyan	1	1
Caribbean Current	2	13	Cyan	1	1
Electric blue	3	6	Cyan	1	1
Teal	4	14	Cyan	1	1
Charcoal	5	4	Black	2	0
Onyx	6	11	Black	2	0
Outer Space	7	9	Black	2	0
Jet Black	8	15	Black	2	0
Canary	9	5	Yellow	3	1
Lemon chiffon	10	7	Yellow	3	1
Xanthic	11	2	Yellow	3	1
Chartreuse	12	16	Yellow	3	1
English Violet	13	10	Magenta	4	1
Amaranth purple	14	3	Magenta	4	1
Orchid	15	12	Magenta	4	1
Plum	16	8	Magenta	4	1

Figure 2: The Designed Motivation Example for Visualization Problems

In the first Column, Categories (colors) were mapped with the numbers from 1 to 16 in order, while in the second column, bijective mapping was done randomly between 1 to 16. The third column was formed according to the main color of our categories. For those categories semantically grouping was performed. For example, the color of 'Pine Green' is the shape of Cyan, while the 'Outer Space' is actually shades of Black. According to these main colors, mapping is done from 1 to 4 for each new category. Categories mapped as Cyan, Black, Yellow, and Magenta, respectively. This semantic grouping may reveal the pattern in the sequence. For example, the DNA strand, which will be used in the statistical model comparison, can be expressed according to the nucleobases; Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). But the same DNA strand can be also mapped in a binary way according to their chemical ring (simple cycle of atoms and bonds). While Adenine (A) and Thymine (T) are grouped as purine bases, Cytosine (C) and Guanine (G) may be represented as pyrimidine bases. In the last column, the mapping for color categories was done in a binary way. For the CMY colors, the number '1' is assigned, while the mixture of colors 'Key', also known as Black, is mapped with 0.

The reason for using four different mappings is to mention several problems of visualization in Categorical Time Series at the same time. The figures will demonstrate how the representation of the same sequence or time series could vary by using different mappings. Thus, this will clearly express the misleading feature of the absence of natural order in categorical variables.

In our motivation example, there is a printing press that printout monochromous banners in 16 different colors. The sequence of printing starts from Pine Green and ends in Plum. Pine green, Caribbean Current, Electric blue, Teal, Charcoal, Onyx, Outer Space, Jet Black, Canary, Lemon chiffon, Xanthic, Chartreuse, English Violet, Amaranth purple, Orchid, Plum

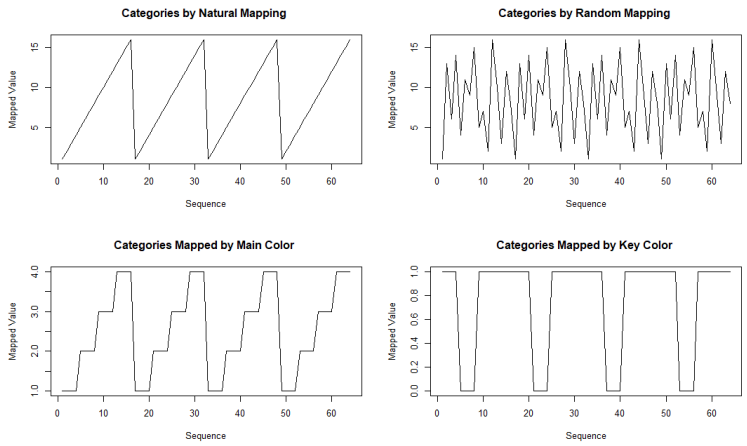


Figure 3: Patterns after printing-out four times

Although the sequence is the same for all graphs -Pine Green to Plum four times-, patterns will be vary and misleading.

In Figure 1, the oscillations in the time series can be interpreted in the wrong way due to the mapping of nominal data types. Even though the general repeating pattern may be observable in graphs, that pattern might become unrecognizable. Especially when there is an increase in the number of categories and the number of transitions between them. The second graph is exemplifying this at an elementary level, by introducing random mapping.

In order to prevent the potential analysis error that may occur after visualization, this paper touches on two different analysis methods and their respective visualizations. The first analysis is the Spectral Envelope proposed by Stoffer, which can be used to detect Periodical behaviors in the categorical data set. The second analysis is the Rate Evolution Graph proposed by Ribler, which is extremely helpful to understand whether a Categorical Time Series has Stationarity or not. The applications and the related visualizations of both those analysis methods on our dataset will be presented in the following subsections.

2 Categorical Time Series Analysis

2.1 Spectral Envelope and Periodicity Analysis

The first analysis, namely Spectral Envelope, treats the time series by not its time dependence but its frequency. The idea might be shortly defined as the implementation of the Fourier Transform to Time Series and computing the spectral density or a sample version of it for given time series data.[23]

2.1.1 Fourier Transformation

Fourier Transformation helps us to understand and analyze the signals or in general perspective any kind of wave by decomposing it into the many sine and cosine wave components, that have different frequencies and amplitudes. With the help of this transformation, the characteristic or key features of the signal can be identified and processing will become easier.

The ground-breaking idea of Fourier Transformation can be defined as the infinite sum or superposition of the individual sine and cosine waves of continuous periodic functions in order to reach any desired function. Fields of usage are generally electric signals, more precisely, in all areas where the periodic waves are present. For example electrical and electronic engineering, signal processing, vibration analysis in mechanical engineering, fluid mechanics, and finally quantum mechanics in Physics.

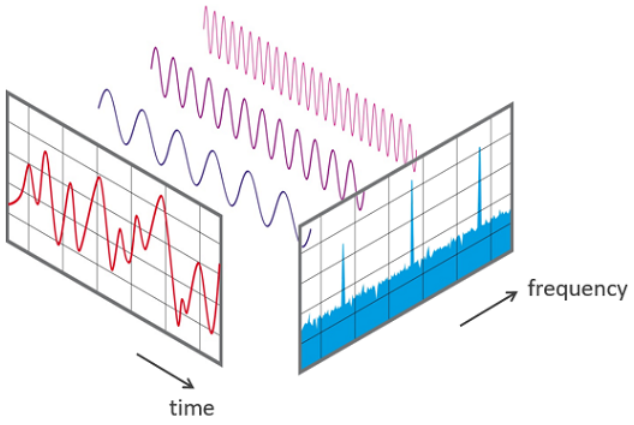


Figure 4: Decomposition by Fourier Transform and Frequency-Domain

2.1.2 The Continuous Fourier Transform

By the help of this formula, any pattern in space and time can be considered a superposition of sinusoidal patterns with different frequencies.[20]

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\xi\pi x} dx$$

The formula shows the transformation of f(x) function
$\hat{f}$ = Fourier transform of Frequency(ξ)
Function of others

2.1.3 The Discrete Fourier Series

$$f(x)\ \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right]$$

$$a_n = \frac{1}{L} \int_{-L}^{L} f(x) \cos \frac{n\pi x}{L} dx, \qquad n = 0, 1, 2, ...$$

$$b_n = \frac{1}{L} \int_{-L}^{L} f(x) \sin \frac{n\pi x}{L} dx, \qquad n = 1, 2, 3, ...$$

A_n and B_n represent the amplitudes of the cosine and sine waves. Since the term $n\pi x$ depends on b, as n changes, the frequencies of the waves will also change.

2.1.4 Spectral Analysis and Periodogram

After the decomposition of the main signal into several signals by the Fourier Transform, the characteristic of each signal can be expressed by its Period and Amplitude. The Period means the horizontal length of one wave, while the Amplitude means depth or vertical length. Among these new signals, one can easily find out the dominant frequency of the main signal. Thus, it will help to find the main periodical pattern in the signal. The visualization of this graph reveals the powerful periodic components, called Periodogram. The sudden over-shoot in the periodogram, the highest magnitude in the spectral density curve, defines the dominant periodical pattern.

At a glance, Spectral Analysis can be described as an analysis of frequency in stationary time series on the frequency domain instead of the time domain. This analysis uses the spectral density function, which takes the covariance function values as the Fourier coefficients. Thus, the function of Spectral Density is described as the Fourier Transform of the autocovariance function.[12]

2.1.5 The Spectral Envelope and Respective Scalings

The first concept of the Spectral Envelope of a Categorical time series was introduced by Stoffer, Tyler, and McDougall in 1993. The main intention was to find a statistical basis for analyzing categorical time series by frequency domain.[21]

In a nutshell, it is the analyzing of the categorical time series, on the frequency domain plane that is formed by Fourier transformation.

To make the concept of the Spectral Envelope and its main idea more clear, the implementation was first done on the motivating example of a color model. After that, the analysis and evaluation of our DNA Sequence dataset are presented.

The methodology is presented below: For a time series, categories are defined as c1,c2, ..., ck. Under the assumption of the time series is stationary, the scaling for each category is $\beta 1, \beta 2 \ldots, \beta k$. The major goal is to find the appropriate scaling of the categories and reveal the spectral information. To do that maximum power – the overshoot in the periodogram – will be considered. [22] At the end of the day, the largest proportion of the variance $\lambda(w)$ is called spectral envelope and it will be described by the frequency w, by introducing respective optimal scaling $\beta(w)$.

$$\lambda(w) = sup_{\beta} \left\{ \frac{f(w;\beta)}{\sigma^2(\beta)} \right\}$$

For the implementation, we have used the R Programming Language and the astsa package. The tool of Spectral Envelope is developed by Stoffer himself and Nicky Poison, according to the articles mentioned in this paper. We have imported the time series as a 1-D Array containing the first 4.000 nucleotides of our dataset in string form. For the smoothing parameter left as NULL, but the default significance of 1e-04 is also used. As an output, the frequencies of the power spectrum, the values of the spectral envelope for each frequency, and the optimal scaling for each frequency point are generated. One of the scalings will be equal to zero. Because when the DNA Sequence processing, each nucleotide (category) is transformed into the 3x1 vector with binary mappings. According to Stoffer's article, this has been done in lexicographical order. While is Thymine transformed into Yt = (0,0,0)', the others A= (1,0,0)', C = (0,1,0)', G = (0,0,1)'.

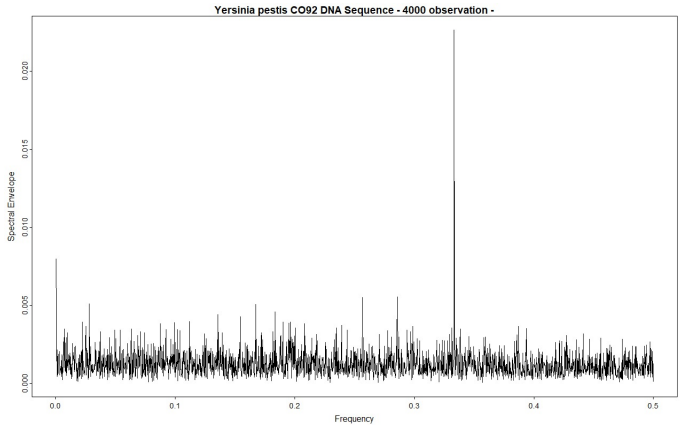


Figure 5: First four-thousand observation of dataset

The Spectral Envelope of the Yersinia pestis virus is demonstrated above. As it can be clearly seen, there is a powerful overshoot in the frequency of 1/3. The position of the peak is identified as 0.3325, while the spectral envelope is maximized (0.023) according to the results. And the selected optimal scaling is assigned as follows:

	Frequency	Spec. Env.	Scaling 'A'	Scaling 'C'	Scaling 'G'	Scaling 'T'
All Observations	0.3332	0.0226	0.45	0.4	0.8	0

Figure 6: Proposed Optimal Scalings

As discussed, the reason for zero scaling for Thymine is due to coding it as the vector of (0,0,0)'. After finding the scaling of the categories, all respective results are shown below. We have also divided our dataset of 4,000 observations into four parts, thousand by thousand. A similar analysis is also presented in Stoffer's article. After we analyze each thousand observations, we can predict whether the respective section with a thousand observations of a nucleic acid molecule is directing the production of a peptide sequence or not. This will provide insight into the coding part of the peptide sequence.

	Frequency	Spec. Env.	Scaling 'A'	Scaling 'C'	Scaling 'G'	Scaling 'T'
All Observations	0.3332	0.0226	0.45	0.4	0.8	0
First Thousand	0.0920	0.0249	0.37	-0.75	-0.55	0
Second Thousand	0.2860	0.0184	-0.54	-0.56	-0.63	0
Third Thousand	0.3330	0.0245	-0.46	-0.47	-0.75	0
Fourth Thousand	0.3330	0.0308	-0.44	-0.53	-0.72	0

Figure 7: Statistics of the each thousand observations

As it can be seen in the below graph and able to supported by the statistics above, the third and fourth thousand observation has very similar characteristics

to the overall sequence. However, the second thousand carry similar features, while the first thousand observations can be identified as non-coding.

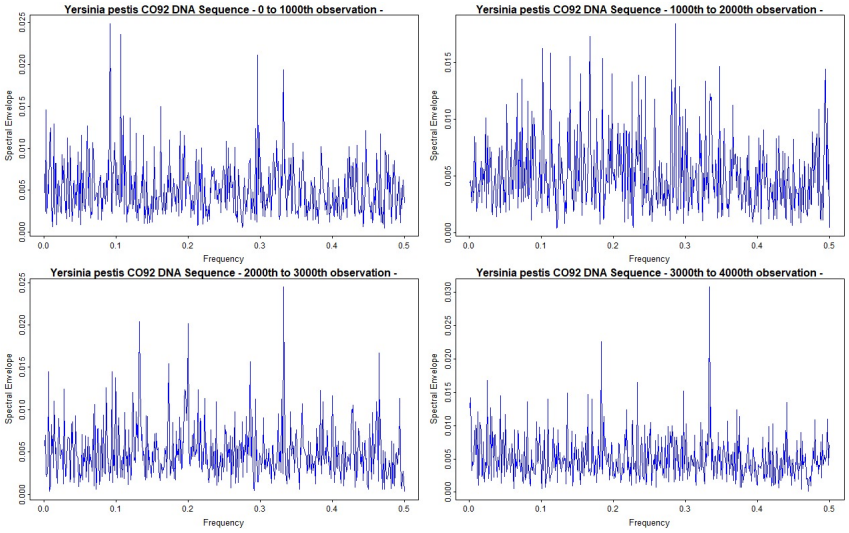


Figure 8: Periodogram of each thousand observation

In the first analysis, the periodicity of the time series was tried to be understood and the Spectral Envelope results were interpreted. Accordingly, the respective DNA part with the first thousand observations was detected as different from the other parts and was determined as the non-coding sequence. In the upcoming analysis, the Stationarity of this non-coding first thousand observations will be assessed. After the Stationarity controls, if the analysis result is, appropriate, Categorical Time Series Statistical models will be employed on this thousand observations in order to predict the nucleotide in this part of a DNA.

2.2 The Rate Evolution Graph and Stationarity Analysis

Rate Evolution Graph is a very simple but surprisingly effective and convenient method in terms of both implementation and evaluation. The result of the Rate Evolution Graph is very straightforward interpretable and provides useful insight into the time series. The method was proposed by Ribler [19] in order to check the stationarity of the time series by visualizing.

The main idea is a visualization of the accumulated sums of each component against time, after the binarization of time-series. The numerical encoding of the classes and the binarization of the time-series allows us to calculate the Rate Evolution Graph. After that, the linearity of the regression line and the slope will provide an understanding about the time-series. The time series can be called stationarity if the rate of evolution develops roughly linearly. Plotting

the rates of evolution of each category on the same graph is one of the most informative methods for fast and simple visual inspection.

For example: Let's consider an analysis of the time series that consists of only a single category of one and has a total of 20 observations.

Time Series: {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}

Obviously, this time series is completely stationary since the properties (category type) have always the same value of 1 at which it has been observed. Rate Evolution Graph results in an array in the following way. Since it can be observed in the graph, the line is completely linear and the slope is one. Thus, the time series has strict stationarity.

1-element Vector:

[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0]

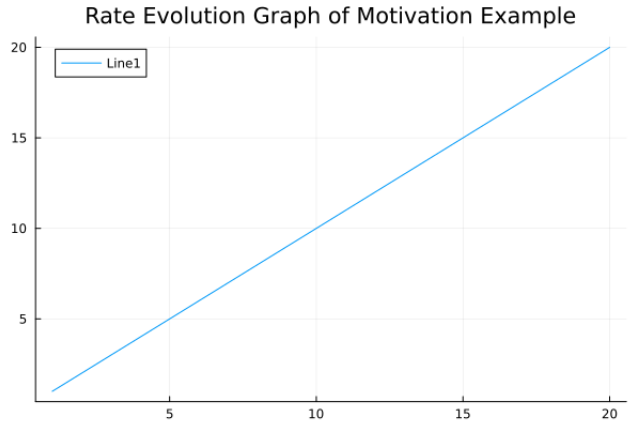


Figure 9: Strict-Stationarity Motivation Example

In order to see the interpretation power of the Rate Evolution Graph, the second category observation has been added to observation points 7,8,15 and 16. Following vector and the respective graph is presented below.

2-element Vector:

[1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 6.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 12.0, 12.0, 13.0, 14.0, 15.0, 16.0]

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 3.0, 4.0, 4.0, 4.0, 4.0, 4.0]

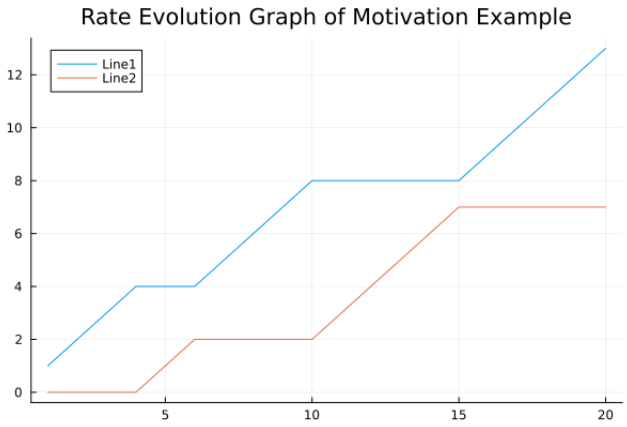


Figure 10: Stationarity Rate Evolution Motivation Example

As it might be seen, the overall linearity of the first category is a bit altered where the second category is observed. This limited-observation-sized exemplification was given by us just for being the motivation example. When the real and very long time-series analysis is considered, straight lines with sharp lines of the graph (e.g., observations 7-8-15 and 16) will be smoother. But one fact remains: The level of convergence to the linearity expresses the level of stationarity while the violations of linearity mean non-stationary.

In order to understand the stationarity of our dataset before employing statistical models, we will check the dataset with Rate Evolution Graph. Below the observed frequencies in the first thousand observations of the DNA Dataset are presented.

Nucleotide	Frequency
Adenine	260
Guanine	250
Cytosine	258
Thymine	232
Total	1000

Table 1: Nucleotides and respective Frequencies

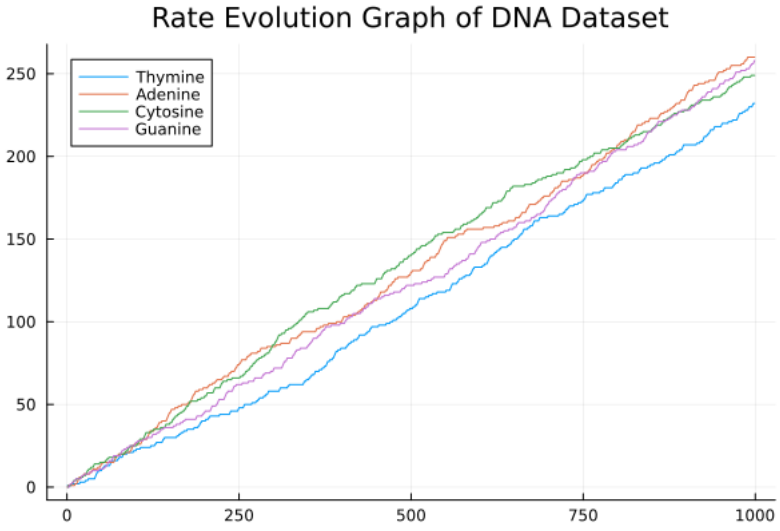


Figure 11: Stationarity Rate Evolution Graph of DNA Sequence

```
4-element Vector{Any}:
 [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 2.0, 2.0 … 229.0, 230.0, 230.0, 230.0, 230.0, 230.0, 231.0, 232.0, 232.0, 232.0]
 [0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0 … 260.0, 260.0, 260.0, 260.0, 260.0, 260.0, 260.0, 260.0, 260.0, 260.0]
 [0.0, 0.0, 1.0, 1.0, 1.0, 2.0, 2.0, 2.0, 3.0, 4.0 … 248.0, 248.0, 248.0, 248.0, 249.0, 249.0, 249.0, 249.0, 249.0, 249.0]
 [0.0, 0.0, 0.0, 1.0, 2.0, 2.0, 2.0, 3.0, 3.0, 3.0 … 253.0, 253.0, 254.0, 255.0, 255.0, 256.0, 256.0, 256.0, 257.0, 258.0]
```

Figure 12: Rate Evolution Vectors of DNA Sequence

In summary, the Rate Evolution Graph counts the cumulative number of each category from the beginning to the end of the time series and displays it as a time-dependent graphic. It has been used to assess the stationary of the DNA sequence. Based on the graphics, it may be concluded that there is a stationary behavior in our DNA Dataset since all four nucleotides have more or less linear characteristics. Thus, after the periodicity and spectral envelope analysis, the second important criterion of the time-series analysis has been satisfied. This allows us to find a reasonable base in order to employ the statistical models on the DNA dataset.

Furthermore, there are also several applications of the non-linear rate evolution graph that enrich the literature. One of the important papers on this topic was written by Brenčič et al. about the Relation between the isotopic composition of precipitation and atmospheric circulation patterns in 2015.

3 Statistical models

3.1 Markov models

3.1.1 Introduction

Models based on the Markov property, as defined by the Russian mathematician Andrey Markov in 1906, are referred to as Markov models. In short, the prediction of an outcome is based solely on the current state's information and not on the previous sequence of events. It is a statistical model that represents the probability of transitioning from one state to another. Markov chain, Markov decision process, hidden Markov model, and partially observable Markov decision process are the four primary types of Markov models.

Let us try to understand Markov Models with the help of an example. In the case of DNA sequences, we can represent each state as a nucleotide (A, C, G, or T), and the transition probabilities represent the probability of transitioning from one nucleotide to another.

To build a Markov model in Python for DNA sequences, we will use the following steps:

1. Read the DNA sequence data from a file or generate a synthetic sequence.
2. Calculate the transition probabilities between each pair of nucleotides.
3. Store the transition probabilities in a transition matrix.
4. Use the transition matrix to generate a synthetic DNA sequence.

The specific applications of each of these models depend on two variables: whether the system state is fully observable and whether the system is controlled or autonomous. A machine learning algorithm can use Markov models to predict the outcome of decision-making processes. A Markov chain may be used to model the outcome if the process is entirely autonomous, meaning there is no feedback that could influence it.

Markov models are basically used widely for clustering and classifying categorical sequences due to their inherent ability to capture complex chronological dependencies hidden within sequential data. Existing Markov models are predicated on the implicit assumption that the probability of the subsequent state depends on the preceding context/pattern, which consists of successive states. This restriction hinders the models because some patterns that are disrupted by noise may not be frequent enough in a sequential form, but may be frequent in a sparse form, which cannot utilize the information concealed in sequential data[4].

3.1.2 Markov Chain Model

The simplest type of Markov model, Markov chains are used to represent systems in which all states are observable. Between states, Markov chains display

the transition rate, which is the probability of transitioning from one state to another per unit of time. This type of model is useful for predicting market crashes, speech recognition, and search engine algorithms, among other applications[3].

Probabilities are modeled with Markov chains using information encoded in the current state. A transition from one state to another occurs stochastically, or with a degree of randomness. Each state has a certain probability of transitioning to each other state, so a Markov chain can predict outcomes based on prior probability data whenever you are in a state and wish to transition[6]. Technically, information is placed in a matrix and a vector - also known as a column matrix - and, after many iterations, Markov chains are composed of a collection of probability vectors. To determine the transition probabilities, a Markov Chain must be "trained" on an input corpus.

Overall, a Markov chain is a mathematical system that undergoes state transitions in accordance with specific probabilistic rules. The defining characteristic of a Markov chain is that the possible future states are fixed, regardless of how the process has reached its current state. In other words, the probability of transitioning to a specific state is solely determined by the current state and the elapsed time. State space, or the set of all possible states, can be anything, including letters, numbers, weather conditions, baseball scores, or stock performance. Markov chains are able to be modeled by finite state machines, and various random walks that are an abundant illustration of their mathematical utility[11].

They are used extensively in economics, game theory, queuing (communication) theory, genetics, and finance, as well as in statistical and information-theoretic contexts. Although it is possible to discuss Markov chains with any size of state space, the initial theory and the majority of applications focus on situations with a finite (or countably infinite) number of states.[11]

We consider the use case of genetic sequence applied to the Markov chain. As described in earlier sections, there are 4 states: A, C, G, and T. The state diagram below depicts the likelihood of the next occurring state starting from any random state. For instance, the Markov chain indicates that there is a 0.184 or 18.4 percent chance that the state T will occur if the current state is G. The state diagram also includes self-loops, wherever needed, to show the occurrence of the same next state.

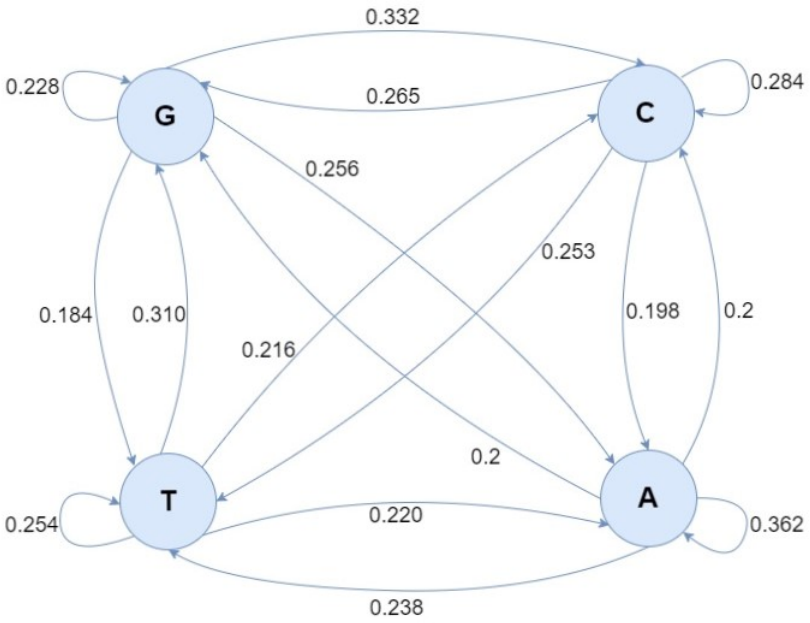


Figure 13: DNA State Diagram- Markov Chain model

The above can also be represented by using a transition matrix which consists of transition probabilities between all the states, i.e., G, C, T and A as below:

	G	C	T	A
G	0.228	0.332	0.184	0.256
C	0.265	0.284	0.253	0.198
T	0.310	0.216	0.254	0.220
A	0.2	0.2	0.238	0.362

Figure 14: DNA transition matrix

The Markov Chain model implemented above leads to the following results: Fig. 3 depicts the frequencies of nucleotides, each in actual and predicted DNA sequence. It can be clearly seen that the count of nucleotides is identical for

'G', whereas it decreases with 'A', 'T', and 'C' respectively.

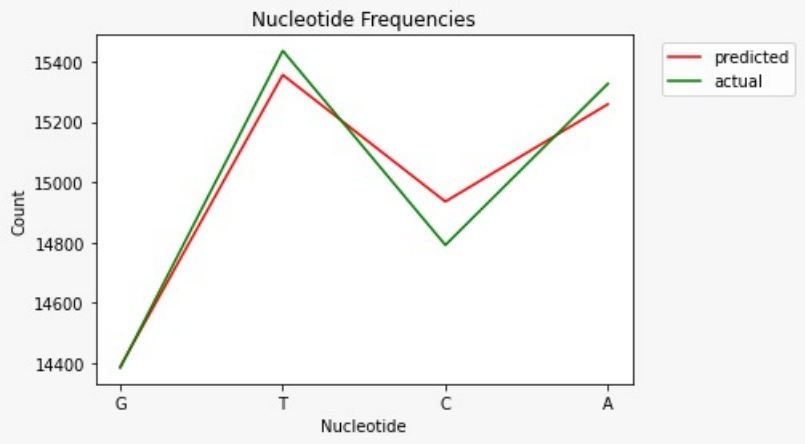
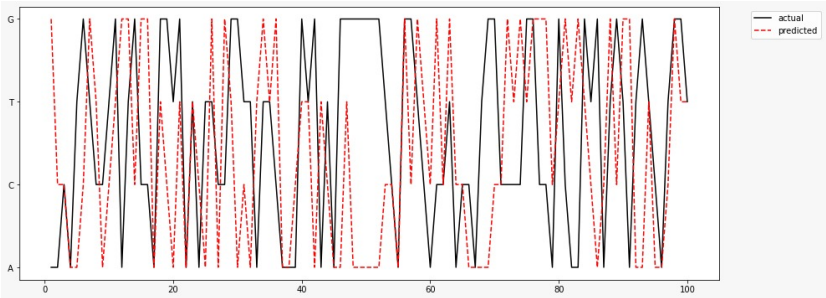


Figure 15: Count of Nucleotides in Predicted and Actual sequence

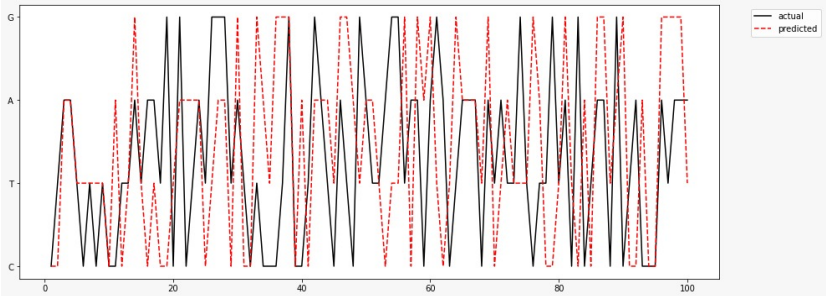
Next, we try to understand the character-wise comparison between the actual and predicted sequence. Since the DNA sequence is too long to be entirely visualized, we randomly pick 4 subsets out of the sequence, containing 100 characters each, and visualize them in each of the 4 graphs below.

This helps to understand the points in the sequence in which conflict occurs between the actual and predicted sequence. Also, we can see a few points where the predicted sequence matches the actual sequence.

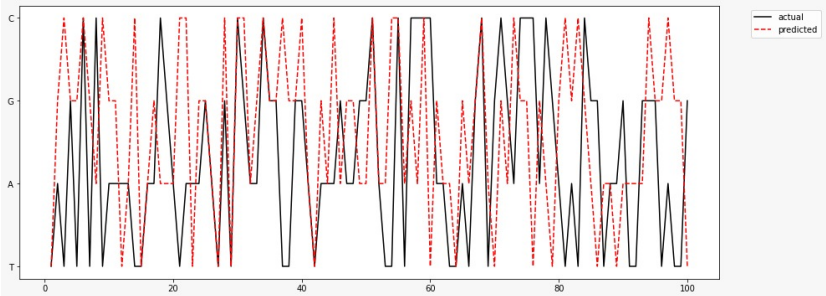
Here, Y-axis denotes each of the nucleotides, i.e., 'G', 'C', 'T', and 'A', whereas X-axis simply denotes the instance of the sequence. For example, in Fig. 4: 1a, the 20th instance in the actual nucleotide is 'T', and in the predicted nucleotide, it is 'A', causing a conflict in this particular instance.



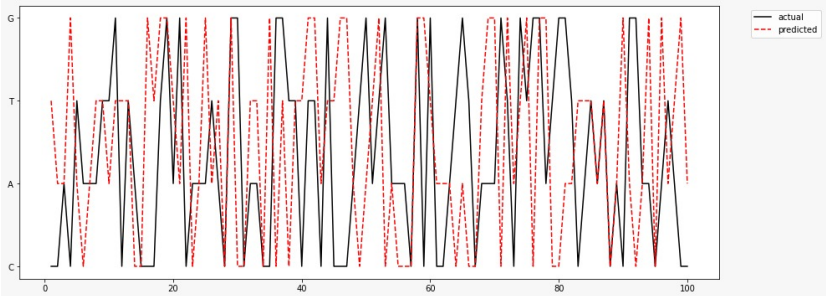
DNA sequence: set1



DNA sequence: set2



DNA sequence: set3



DNA sequence: set4

Figure 12: Comparison of actual and predicted DNA sequence

Concluding remarks: In summary, we implemented Markov Chain Model on DNA data sequence. First, a transition matrix is formed based on the transition probabilities between the nucleotides. Based on those, a synthetic sequence is predicted. We compare the actual and predicted sequence in terms of the frequencies of each nucleotide, as well as the character-wise comparison between the sequences. For this purpose, we randomly pick subsets out of the data but observe approximately consistent performance on each of the subsets.

We attempt to increase the order of the transitions, which leads to better performance. For instance, consider model A with a length of sequence 100 and order 3. On the other hand, consider another model B with a length of 1000 and order 10. Model B yields much better performance as compared to model A.

However, the model does not provide very promising results. In the next section, we study another type of model, i.e., Hidden Markov Models.

3.1.3 Hidden Markov Model

The statistical model HMM, or Hidden Markov Model, is used to assess sequential data in fields including bioinformatics, speech recognition, natural language processing, etc.

An HMM models the system as a set of observations (emissions) obtained from a series of hidden states. The hidden states cannot be seen explicitly, but they affect the observed emissions. In other words, HMMs simulate systems that generate observed emissions by using hidden states. In a process known as pattern theory, a series of output tokens will provide insight into the sequence of state transitions[6].

The model is defined by the probabilities of starting in each hidden state (Starting or Initial state probability), transitioning between hidden states (Transition probability), and emitting each observation out of each hidden state (Emission probability). Along with the transition probabilities as in Markov Chain model, HMMs make use of Emission probabilities in order to predict the hidden states for given observations.

Let us understand HMMs with our DNA example. In the below figure, we can see a generalized structure of HMM on DNA data sequence.

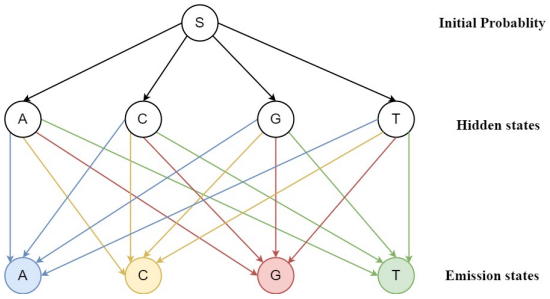


Figure 13: DNA State Diagram-HMM

In the above figure, the top layer node 'S' represents the initial state. The middle layer denotes hidden states, which will be predicted based on the emission states in the bottom layer. Each hidden state is dependent on each of the emission probabilities associated with it. The emission probabilities as well as the transition probabilities separately sum up to 1. Also, each of the emission states is calculated based on the previous hidden state.

We implement the Hidden Markov Model on the same use case, that is, DNA sequence data, which was also used for Markov Chain Models. Here, we consider 4 emission states: 'A', 'C', 'G', and 'T'. We consider n no. of instances. Here, our DNA sequence consists of n=59940 instances. For fitting the Categorical HMM model, we pass the emission states (components) along with some random state. Based on the emission states, we find hidden states for each of these instances. Thus we fit the model on DNA data sequence (emissions). We numerically encode the nucleotides for the sake of fitting the model and decode them back in order to understand the model visually.

Mathematically, the task of HMM is to maximize the probability of all the variables occurring together, that is, find the combination of hidden states such that the probability is maximized.

$$\max_{A_H, C_H, G_H, T_H} \mathbb{P}(NE_1 =' A', NE_2 =' G', NE_3 =' T', ..., NE_{59940} =' C', \\ NH_1 =' A'_H, NH_2 =' C'_H, NH_3 =' G'_H, NH_4 =' T'_H)$$

The transition matrix and emission matrix can be extracted from the fitted HMM model and are as shown below:

```
trans_mat = model.transmat_
print(trans_mat)
```

```
[[1.77766725e-02 6.34484872e-02 3.72777228e-02 8.81497117e-01]
 [8.52736022e-01 1.41237697e-03 6.09319162e-02 8.49196844e-02]
 [2.50257597e-03 1.80910586e-01 1.16406899e-01 7.00179939e-01]
 [4.03283886e-04 1.01751280e-01 4.49420329e-01 4.48425107e-01]]
```

Figure 14: Transition matrix- HMM

```
emission_prob=model.emissionprob_
print(emission_prob)
```

```
[[0.315323   0.25138822 0.19494798 0.23834079]
 [0.13416533 0.27575604 0.20851008 0.38156855]
 [0.25840727 0.25148808 0.15940006 0.33070458]
 [0.27131035 0.22308308 0.3296442  0.17596237]]
```

Figure 15: Emission matrix- HMM

In the next part, we try to evaluate the overall performance of our model. Thus, we split the entire dataset into training part (80%) and testing part (20%). We train our model on the training part, and later predict the sequence on testing part. However, the true labels for the DNA data are not available. For the purpose of calculating accuracy, we create dummy labels, first as all zeroes, and later as all ones. In this case, the accuracy of the model is not very promising, which is obvious because of the dummy labels. It is highly biased on random state as well as the chosen true probabilities.

The primary benefit of this model is the recovery of a hidden data sequence by observing an output dependent on the hidden data sequence[5]. These are used to represent systems with some states that are not observable. In addition to displaying states and transition rates, hidden Markov models display observations and likelihoods of observations for each state[3].

However, with the emergence of modern computers, this drawback was reduced, and hidden Markov models quickly gained prominence as a tool for supervised machine learning. Speech recognition was one of the earliest applications of HMMs. In reference to the classic tutorial based on HMMs by Rabiner (1989), the fundamental theory of HMMs was outlined by Baum and colleagues (Baum and Petrie, 1966) in the late 1960s and subsequently developed by various groups in the 1970s. In one of his earliest reviews on the topic, Eddy (1996) discusses the acceptance of HMMs in biology[2].

3.2 Discrete NDARMA models

3.2.1 Literature Review

As discussed in the previous section, Markov chains have been widely used as models for stationary discrete time series. However, they are over-parametrized for statistical purposes [13]. Further, the problem arises when the data to be modelled is non-Markovian, or not first-order Markovian. In case Higher-order Markov chains are used, this only leads to the problem of over-parametrization.

Let us now review the history of NDARMA models for discrete time-series data. The theory of ARMA models (Box & Jenkins, 1976) has been playing a vital role in the modelling of continuous time series data which arises from

several practical fields. Later in 1983, Jacobs & Lewis proposed the 'new' discrete ARMA models. Conventional ARMA models could be easily adapted for categorical data, by offering some kind of counterpart [23]. Over the last couple of decades, research has been continued in formulating discrete ARMA models in different ways.

An alternative method of the formulation was demonstrated by Biswas & Song (2009). It extended the use of Pegram's operator to define discrete-valued ARMA processes, which was originally proposed only for AR processes, i.e. for the modelling of continuous time series data. The proposed model is able to analyze any type of discrete data. It is applied to a number of real datasets, for instance, a small sample from Infant sleep data by Stoffer et al. (1988). Analyses showed that the performance of these models was slightly better for certain cases.

3.2.2 Various methods used for Discrete NDARMA models

1. **Backshift mechanism**

   The central idea of Jacob & Lewis's work is to ultimately generate an ARMA-like dependence structure through a certain random mixture [23]. It was formulated using the backshift mechanism as follows The research is based on the assumption that NDARMA and DARMA processes are real-valued. They can, however, be applied to model categorical time series, in which case the numerical measures of evaluation would not be applicable.

   A discrete time series with the correlation structure of a mixed moving average autoregressive process is defined. This new process resembles the linear ARMA (p, N) process. Following is the key idea that leads to this particular model, consider a probabilistic mixture of a finite number of random variables. Each of these variables has probability mass function π, and probability mass function π even if they are dependent.

   Thus, for p = 1,2,. . . and N = 0, 1,2,. . . let

   $$X'_n = V_n X'_{n-A_n} + (1 - V_n) Y_{n-D_{n'}}$$

   where $\{V_n\}, \{A_n\} and \{D_n\}$ are as before. Thus, with probability ρ, X'_n is one of the p previous values X'_{n-1},. . . , X'_{n-p} and with probability (1 - ρ) it is a mixture of the previous Y_k s, n - N $\leq$ k $\leq$ n.

   Let $\tau = \inf \{i : \delta_i > 0\}$. Note that

   $$Z'_n = \{(X'_n, X'_{n-1}, ..., X'_{n-p+1}, Y_{n-\tau}, ..., Y_{n-N}), n = 1, 2, ...\}$$

   is a Markov Chain with state space F which is equal to the product space of E with itself ρ + (N - τ + 1) times. Since

   $$P\{X'_{n+\tau+1} = Y_{n+1} = j | X'_0, ..., X'_n, Y_0, ..., Y_n\} \geq (1 - \rho)\delta_\tau \pi(j),$$

there is a set J $\subset$ F such that

$$\min_{i \in F, k \in J} P\{Z'_{n+K} = k | Z'_n = i\} = \gamma > 0,$$

where

$$K = p + N$$

and

$$J \subset \{X'_{n+K} = Y_{n+K-\tau}, X'_{n+K-1-\tau}, ..., X'_{n+K-p+1} = Y_{n+K-p+1-\tau}\}$$

2. **Pegram's mixing operator**

Another implementation uses Pegram's operator to define discrete-valued time series.

For a given coefficient $\phi \in$ (0,1), and two independent discrete random variables U and V Pegram's operator mixes them so that a random variable Z is generated as follows

Z: Z = (U, ϕ) * (V, 1- ϕ)

In this case, the marginal probability function is given by

P(Z=j) = ϕ P(U=j) + (1-ϕ) P(V=j), j=0,1,...

Hence, Pegram's operator produces a mixture of two discrete distributions, with the respective mixing weights as ϕ and (1-ϕ).

3. **Extension to ARMA**

Let us see another implementation by Weiß & Göb (2008). Assume that the innovations $(\epsilon_t)_Z$ and the observations $(X_t)_Z$ are categorical processes with state space S, where $(\epsilon_t)_Z$ is independent and identically distributed with marginal distribution π, and ϵ_t is independent of $(X_s)_{s<t}$. The random mixture is obtained through the multinomial random vectors (Decision Variables.[24])

$$(\alpha_{t,1}, ..., \alpha_{t,p}, \beta_{t,0}, ..., \beta_{t,q}) \sim MULT(1; \phi_1, ..., \phi_p, \varphi_0, ..., \varphi_q),$$

These vectors are independent of $(\epsilon_t)_Z$ and $(X_s)_{s<t}$. Then, $(X_t)_Z$ is defined as NDARMA(p, q) process.

3.2.3 Results

The performance of ARMA models was found to be particularly good through simulation studies. It matched the performance of maximum likelihood estimators for the first-order autoregressive case [13]. Moreover, these are much simpler for computation than the maximum likelihood estimators. Specifically, Pegram's AR models are observed to be much more flexible in terms of ease of interpretation and the range of correlation. Therefore, they established a unified framework of discrete-valued stationary processes. Analyses showed that for particular cases where $Y_{t-1} = Y_t$ is considerably frequent, the performance of these models was slightly better than the conventional models.

3.3 Other Regression Models

This section includes the regression models for time series in categorical data. The advantage of the regression model is that it embraces independent variable details and let represented by vector format (Z) The regression models included in this paper is based on Generalized linear models (GLM). The known information is conditionally linked to present the next outcome hence a few of the last observations are needed to predict the current or future observation. The data that we use demonstrate our topic will use for both covariates and the observation. The concept of regression from paper [14] depicts here,

If it assume a time series with m categories,

$$\{Y_t, t = 1, 2, 3, \ldots.\}$$

The t^{th} observation can be declared as a vector of,

$$[Y_t = Y_{t1}, Y_{t2}, ..., Y_{t_{(m-1)}}]'$$

which $Y_{tj} = 1$ if j^{th} category observed otherwise 0 for other observation. Let vector of conditional probabilities given by,

$$\pi_t = (\pi_{t1}, \pi_{t2}, ..., \pi_{t(m-1)})'$$

where,

$$\pi_{tj} = P(Y_{tj} = 1 | Y_{t-1}, Y_{t-2}, \ldots\ldots, Y_1)$$

Probability of occurring any observation out of m categories can be determine by finite observation

3.3.1 Link function

Link function $\log\left(\frac{\pi}{1-\pi}\right)$ is a key concept used to relate a response variable to one or more predictor variables and the only change from the linear regression is the logit function which is the log of odd of success. These log odds are a

function of π and it's a function of the mean of 1 and 0 and odds range from 0 to infinity. Logistic regression is a type of general generalized linear model and all of them have a link function which is the means of Y hence the purpose is not to directly models the values of Y but the function of the mean of Y.

This function transforms the probabilities of the categories of a categorical response variable to a continuous scale that has no bounds. After this transformation, the relationship between the predictors and the response can be modeled using linear regression. For instance, a binary response variable can only have two distinct values. By transforming these values into probabilities, the response variable can range from 0 to 1, which allows for linear regression modeling.

When selecting a link function, it's important to choose one that best fits the data. certain link functions have a special significance in a particular field. For instance, the logit link function has an advantage in that it estimates the odds ratios. For this paper, we will be using the logit link function as our data follows nominal.

3.3.2 Logistic Regression

David Cox [25] a statistician, first described the method of binary logistic regression. However, the idea of logistic regression can be traced back to the early 19^{th} century, when mathematician Pierre-Simon Laplace introduced the "logistic function". In binary logistic regression, it is assumed that the connection between the log-odds of the binary outcome and the predictor variables is linear [1](section 4.2)

$$\log\left(\frac{\pi_x}{1-\pi_x}\right) = \beta_0 + \beta X$$

Binary logistic regression requires a dichotomous dependent variable that takes one of two values, usually represented as 1 and 0. The independent variables can be categorical or continuous. The model calculates the coefficient of the independent variables, which signifies the alteration in the log-odds of the dependent variable associated with a unit change in the independent variable while holding all other variables constant. Using these coefficients, it is possible to forecast the likelihood of the dependent variable taking on a particular value, based on the values or categories of the independent variables.

3.3.3 Multinomial Logistic Model

Multinomial logit model introduced by economist Daniel McFadden [18] in 1973 and the intention was to provide a statistical framework for analyzing and predicting the choices individuals make when faced with multiple alternatives. In this paper, we are adopting Agresti [1] section 6.1 methodology for explaining and application approach.

Let J denoted number of categories and let response probabilities denoted by

$\pi_1, \pi_2, \pi_3, ..., \pi_j$ where $\sum \pi_j = 1$

The multinomial probability distribution describes the likelihood of observing J categories among n observations for categorical outcomes. This distribution forms the basis for the multicategory logit model.

In this model, the order of the categories is irrelevant, as it treats the categories as nominal. The method generates a multinomial logit model for all combinations of categories by comparing the odds of an observation being in a particular category to the odds of it being in the reference or base category.

If the last category (J) is chosen as the baseline, the baseline-category logits,

$\log\left(\frac{\pi_j}{\pi_J}\right)$ where j = 1,2,. . . . . ..,j-1.

If the response falls in categories j or J, then the log odds of the response being j is determined by,

$$\log\left(\frac{\pi_1}{\pi_J}\right), \log\left(\frac{\pi_2}{\pi_J}\right), ..., \log\left(\frac{\pi_{J-1}}{\pi_J}\right)$$

For an example: The model in the case where J equals 3, $\log\left(\frac{\pi_1}{\pi_3}\right)$ and $\log\left(\frac{\pi_2}{\pi_3}\right)$. The baseline category logit model [1] (Section 6.1.1),

$$\log\left(\frac{\pi_j}{\pi_J}\right) = \beta_0 + \beta_j X \tag{1}$$

where j = 1,2,.....,J-1

Each of the J categories in the model, it has J-1 equations with distinct parameters. The effect of the parameters differs depending on the category paired with the baseline. If J equals 2 which category logit model $\log\left(\frac{\pi_1}{\pi_2}\right)$ then the model reduces to standard logistic regression for binary responses.

An alternative way to convey the multicategory logit model is in terms of the probabilities of the response [1] (Section 6.1.3).

$$\pi_j = \frac{e^{\alpha_j + \beta_j X}}{\sum_h e^{\alpha_h + \beta_h X}} \tag{2}$$

As this is one of the models used in this paper, let's work through an example. Suppose we want to analyze the factors influencing the choice of transportation to school in Trier. We have a categorical dependent variable consisting of four options: walking, cycling, car, and taking the bus. Additionally, we have a few independent variables, such as distance, parents' income, and age, which we believe may influence the choice of transportation.

We fit a multinomial logit model to the data, which estimates coefficients for each independent variable. These coefficients represent the effect of each independent variable on the log-odds of the dependent variable, which is the choice of transportation. For instance, if the coefficient for income is negative compared to car and bus, this suggests that children of wealthier parents tend to choose the car more frequently than the bus.

Our specific interest lies in papers by Konstantinos Fokianos and Benjamin Kedem [7] for the application of multinomial logit models for categorical time series,

$$\log\left(\frac{\pi_{ti}\beta}{\pi_{t4}\beta}\right) = \beta_{i0} + \beta_{i1}Y_{(t-1)1} + \beta_{i2}Y_{(t-1)2} + \beta_{i3}Y_{(t-1)3}$$

for i = 1, 2, 3 and it is denoted by $1 + Y_{t-1}$. A second- order model is labeled $1 + Y_{t-1} + Y_{t-2}$ and consists of 27 plus a linear combination in terms of $Y_{(t-2)1}, Y_{(t-2)2}, Y_{(t-2)3}$ and so on.

Model 1 $1 + Y_{t-1}$
Model 2 $1 + Y_{t-1} + Y_{t-2}$
Model 3 $1 + Y_{t-1} + Y_{t-2} + Y_{t-3}$

3.3.4 Application of Multinomial logit model

The specific dataset used in this study is comprised of FASTA format sequence data for a bacterial genome with the accession number CP009973.1 , which is maintained by the European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL-EBI). This RNA has 4 nucleotides Thymine (T), Cytosine (C), Adenine (A), and Guanine (G).

R installation includes the nnet package which provides access to the multinom function. This function employs neural networks to fit multinomial logit models [8]. Here we have implemented the model 1st order model [1] for the 1st 1000 data list where the preceding DNA sequence is considered as a response variable and the general DNA list is considered as an independent variable. The encoding for the four nucleotides is given in one number as follows for this model

1 for Guanine , 2 for Adenine, 3 for Cytosine, 4 for Thymine

Model	AIC	BIC
$1 + Y_{t-1}$	2758.319	2817.212
$1 + Y_{t-1} + Y_{t-2}$	2778.248	3013.820
$1 + Y_{t-1} + Y_{t-2} + Y_{t-3}$	2893.180	3835.469

Table 2: AIC and BIC calculation for different Multinomial Logit Models

By observing the table 2, it is evident that the best model is 1 + Y[t-1]. This model outperforms all the others, with the lowest AIC score. The second-best model has an AIC score that is more than 20 units higher than the best model. Given this comparison, we would select the 1 + Y[t-1] model for employing the multinomial logit regression model.

```
Coefficients:
  (Intercept) as.factor(x)2 as.factor(x)3 as.factor(x)4
2   0.1158266     0.4762229    -0.4035119   -0.46066473
3   0.3757923    -0.3757899    -0.2912389   -0.74043284
4  -0.2144081     0.3902951     0.1692923    0.01528008
```

Figure 16: Computer output of coefficient of categorical logit model

Once the coefficients substitute to model,

$$\log\left(\frac{\pi_2}{\pi_1}\right) = 0.1158 + 0.4762Y_{(t-1)2} – 0.4035Y_{(t-1)3} – 0.4606Y_{(t-1)4} \tag{3}$$

$$\log\left(\frac{\pi_3}{\pi_1}\right) = 0.3757 – 0.3757Y_{(t-1)2} – 0.2912Y_{(t-1)3} – 0.7404Y_{(t-1)4} \tag{4}$$

$$\log\left(\frac{\pi_4}{\pi_1}\right) = -0.2144 + 0.3902Y_{(t-1)2} + 0.1692Y_{(t-1)3} + 0.0152Y_{(t-1)4} \tag{5}$$

The Coefficient section presents three models that how effect $Y_{(t-1)2}(adenine)$, $Y_{(t-1)3}(cytosine)$ and $Y_{(t-1)4}(thymine)$ impact the log odds of adenine to guanine, cytosine to guanine, and thymine to guanine.

It is challenging to interpret coefficients in logit models, except for determining whether the coefficient is positive or negative, which is relatively straightforward. For an example, the $Y_{(t-1)2}(adenine)$ coefficient for cytosine is negative (-2.3757). It appears that the $Y_{(t-1)2}(adenine)$ is the less likely to be cytosine instead of Guanine

```
Std. Errors:
  (Intercept) as.factor(x)2 as.factor(x)3 as.factor(x)4
2   0.1821234     0.2510737     0.2597743     0.2581970
3   0.1720233     0.2608706     0.2404401     0.2519542
4   0.1981993     0.2732083     0.2633882     0.2648037
```

Figure 17: Computer output of standard errors for categorical logit model

The standard errors provide a measure of the degree of uncertainty in the estimation of coefficients. For instance, it can be interpreted as the $Y_{(t-1)2}$ (adenine) coefficient for thymine is 0.3902 with a standard error of 0.2732. When

the standard error is relatively, it suggests a higher level of accuracy in the estimate.

```
Value/SE (Wald statistics):
  (Intercept) as.factor(x)2 as.factor(x)3 as.factor(x)4
2   0.6359787      1.896745     -1.553317   -1.78416019
3   2.1845431     -1.440522     -1.211274   -2.93876020
4  -1.0817805      1.428562      0.642748    0.05770343
```

Figure 18: Computer output of Wald test for categorical logit model

The standard Wald statistics displays the ratio between the coefficients and the standard errors. If the ratio is significantly different from 0, it indicates a more accurate estimate is either positive or negative.

Estimating Response Probabilities

The idea of log odds is not inherently comprehensible to humans. In order to grasp the model, it may be necessary to present it in terms of probabilities. The probabilities associated with the four responses:

$$\pi_1 = \frac{1}{1 + e^{0.1158+0.4762Y_{(t-1)2}-0.4035Y_{(t-1)3}-0.4606Y_{(t-1)4}} + e^{0.3757-0.3757Y_{(t-1)2}-0.2912Y_{(t-1)3}-0.7404Y_{(t-1)4}} + e^{-0.2144+0.3902Y_{(t-1)2}+0.1692Y_{(t-1)3}+0.0152Y_{(t-1)4}}} \quad (6)$$

$$\pi_2 = \frac{1 + e^{0.1158+0.4762Y_{(t-1)2}-0.4035Y_{(t-1)3}-0.4606Y_{(t-1)4}}}{1 + e^{0.1158+0.4762Y_{(t-1)2}-0.4035Y_{(t-1)3}-0.4606Y_{(t-1)4}} + e^{0.3757-0.3757Y_{(t-1)2}-0.2912Y_{(t-1)3}-0.7404Y_{(t-1)4}} + e^{-0.2144+0.3902Y_{(t-1)2}+0.1692Y_{(t-1)3}+0.0152Y_{(t-1)4}}} \quad (7)$$

$$\pi_3 = \frac{e^{0.3757-0.3757Y_{(t-1)2}-0.2912Y_{(t-1)3}-0.7404Y_{(t-1)4}}}{1 + e^{0.1158+0.4762Y_{(t-1)2}-0.4035Y_{(t-1)3}-0.4606Y_{(t-1)4}} + e^{0.3757-0.3757Y_{(t-1)2}-0.2912Y_{(t-1)3}-0.7404Y_{(t-1)4}} + e^{-0.2144+0.3902Y_{(t-1)2}+0.1692Y_{(t-1)3}+0.0152Y_{(t-1)4}}} \quad (8)$$

$$\pi_4 = \frac{e^{-0.2144+0.3902Y_{(t-1)2}+0.1692Y_{(t-1)3}+0.0152Y_{(t-1)4}}}{1+e^{0.1158+0.4762Y_{(t-1)2}-0.4035Y_{(t-1)3}-0.4606Y_{(t-1)4}} + e^{0.3757-0.3757Y_{(t-1)2}-0.2912Y_{(t-1)3}-0.7404Y_{(t-1)4}} + e^{-0.2144+0.3902Y_{(t-1)2}+0.1692Y_{(t-1)3}+0.0152Y_{(t-1)4}}} \quad (9)$$

independent variable	response			
	G(Y_t)	A(Y_t)	C(Y_t)	T(Y_t)
G($Y_{(t-1)}$)	0.2280	0.2560	0.3319	0.1840
A($Y_{(t-1)}$)	0.2000	0.3615	0.2000	0.2384
C($Y_{(t-1)}$)	0.2635	0.1976	0.2868	0.2519
T($Y_{(t-1)}$)	0.3103	0.2198	0.2155	0.2542

Table 3: Probability Calculation for Categorical Logit Model

Table 3 displays estimated probabilities for the four response categories. To illustrate, let say the observed nucleotide A ($Y_{(t-1)2} = 1$) then eventually $Y_{(t-1)1} = 0, Y_{(t-1)3} = 0$ and $Y_{(t-1)4} = 0$ and the estimated probability of response [2] T equals

$$\pi_4 = \frac{e^{-0.2144+0.3902(1)+0.1692(0)+0.0152(0)}}{1+e^{0.1158+0.4762(1)-0.4035(0)-0.4606(0)} + e^{0.3757-0.3757(1)-0.2912(0)-0.7404(0)} + e^{-0.2144+0.3902(1)+0.1692(0)+0.0152(0)}} \quad (10)$$

According to the model, it is clear that if the observed nucleotide is Guanine, there is a greater probability of it being Cytosine and a lower probability of it being Thymine. It is observed that there is a great deal of uncertainty in predicted probabilities for the next nucleotide when the current nucleotide is Cytosine. Likewise, it can deduce the probabilities of what is probable and improbable from the table.

The model was utilized to forecast the succeeding 10 nucleotides. On average, it accurately predicted three nucleotides, while failing to predict four nucleotides at any instance.

4 Conclusion and Discussion

This paper discusses novel analysis approaches, statistical models, and current literature in Categorical Time Series Analysis. One of the main goals was to understand the theoretical and practical differences between studies and statistical models in the nascent field of Categorical Time Series Analysis. In order to do that models were tested on a previously analyzed real and challenging dataset. Dataset selection was made in parallel with the research in the literature and a version approved by the European Nucleotide Archive was selected.

One of the expectations was to explore which models generate better results under which conditions. Furthermore, we have investigated the advanced further implementations in the literature. Finally, we have compared the theoretical and practical differences between these models in order to understand their strengths and weaknesses.

Before the applications of Statistical Models, some analyses and visualizations were made for the selected data set. The reason for these analyzes is to understand the structure and patterns of the dataset. The information obtained from these analyses will directly affect the predictive power and applicability of statistical models as well as several important decisions such as; model and parameter selection. In this direction, Periodicity and Stationarity analyses, which are two essential concepts in time series analysis, were carried out and the structure of the data set was shed light on. While performing the Periodicity Analysis, one of the most important methods in the literature, Spectral Envelope, was used. For the Stationarity Analysis, the simple but incredibly effective Rate Evolution Method was used. As a result of the Periodicity Analysis, it was observed that the DNA part consisting of the first thousand observations was different from the other parts. It was decided to apply the statistical methods to create predictions on this different part. This also provided a chance of testing models against challenging tasks. As a result of the subsequent Stationary analysis, it was observed that the first thousand observations were Stationary. This means that the application of statistical models is feasible for this part of DNA.

In this paper, not only the difference between the types of statistical models are discussed but also various versions of specific models. For example, several options such as 1st order, 2nd order and 3rd order were inspected in Hidden Markov Model. In order to decide the version of Statistical Models, several criteria and metrics were considered. The main criteria for deciding the version in Multinomial Logit Model was AIC, while it was prediction power and accuracy for Markov Chains.

In summary, implementing an NDARMA model on DNA sequences is quite challenging due to complex correlations, and the large length of these sequences. To model these sequences accurately, sophisticated techniques such as time-varying autoregressive models or Markov methods are better suited. Hence, for the scope of this research, Markov Chains, Hidden Markov Models, and Multinomial Logit models were selected for the implementation of our use case.

We implemented the Markov Chain model on our use case, based on the fundamental building blocks - Nucleotides 'A', 'G', 'C', and 'T' as fixed states and transition probabilities between them. Nucleotide frequencies and character-wise comparisons are used to evaluate the predicted synthetic sequence with respect to the real sequence. Our research demonstrates that improving the order of transitions improves performance. The model does not still yield promising outcomes, which leads us to further research another type of model, i.e., hidden Markov models.

Hidden Markov Models (HMMs) are built upon the basic foundations of Markov Chains, along with other essential components - hidden states and emis-

sion probabilities. We built the Markov Chain Model by training on the selected training part of the DNA dataset and tested it on the remaining part. However, since the real labels for this dataset are not available, we measured accuracy using dummy labels. In this case, the accuracy is not encouraging as it heavily favors the choice of random state and true probability.

In general, we find that Markov Chains perform better when capturing short-term dependencies (low order), whereas, for higher order dependencies, Hidden Markov models are a better choice.

Besides the theoretical comparison of the above-mentioned models, the subsequent 10 nucleotides were also predicted by using both Markov and Hidden Markov models. According to the results, the 3rd-order Hidden Markov Chain model had better accuracy than the other models, consistently performing with higher accuracy every time we executed it. Although the accuracy rate was not very promising, still it gives the chance to compare it with the prediction of the Multinomial Logit Model.

Last but not least, we have created a multinomial logit model for all combinations of categories by comparing the odds of an observation being in a particular category to the odds of it being in the reference category which is Guanine. The best model for multinomial logit regression is found to be $1 + Y_{(t-1)}$, which has the lowest AIC score compared to other models. We were able to generate a probability table where we can predict the next nucleotide based on the current nucleotide. However, the model accuracy was low when we try to predict the upcoming part of the DNA Sequence.

After comparing the Hidden Markov and Multinomial Logit models, we found that the accuracy of the Hidden Markov model was consistently higher than the Multinomial Logit model. Therefore, we concluded that the Hidden Markov model performed better among all the other statistical model for the selected dataset even though the accuracy rate was not totally promising for us.

We have also further investigated the root cause of the accuracy problem. According to our investigation, statistical models were heavily under effected by the uniform- distribution of the dataset. According to the frequency table and stationarity check, which presented in the Analysis Chapter, it has been identified that all nucleotides are existed roughly equally and shows a uniform distribution throughout the selected DNA sequence. This distribution, which is almost equal to each for all nucleotides, reduces the differences between nucleotides and complicates the prediction power. This effect is valid, especially for the models discussed in this paper since they are directly affected by probability distributions. As it might be remembered, this result was already expected as a result of previous visual categorical time series analyses in Table 1. Although it was challenging, the desired research was to test the capabilities of statistical models under such a difficult task and has been performed.

5 Appendix

5.1 Dataset - DNA Sequence of Yersinia pestis

GTAGCCGTCG	TAGCCGTCGG	AGCCGTCGGC	GCCGTCGGCA	CCGTCGGCAC
CGTCGGCACG	GTCGGCACGA	TCGGCACGAA	CGGCACGAAA	GGCACGAAAA
GCACGAAAAT	CACGAAAATG	ACGAAAATGC	CGAAAATGCC	GAAAATGCCA
AAAATGCCAG	AAATGCCAGA	AATGCCAGAC	ATGCCAGACT	TGCCAGACTG
GCCAGACTGG	CCAGACTGGG	CAGACTGGGT	AGACTGGGTG	GACTGGGTGC
ACTGGGTGCA	CTGGGTGCAG	TGGGTGCAGA	GGGTGCAGAC	GGTGCAGACA
GTGCAGACAG	TGCAGACAGG	GCAGACAGGT	CAGACAGGTT	AGACAGGTTT
GACAGGTTTT	ACAGGTTTTA	CAGGTTTTAT	AGGTTTTATC	GGTTTTATCG
GTTTTATCGA	TTTTATCGAA	TTTATCGAAT	TTATCGAATA	TATCGAATAT
ATCGAATATC	TCGAATATCT	CGAATATCTG	GAATATCTGC	AATATCTGCG
ATATCTGCGC	TATCTGCGCC	ATCTGCGCCG	TCTGCGCCGC	CTGCGCCGCT
TGCGCCGCTT	GCGCCGCTTT	CGCCGCTTTC	GCCGCTTTCC	CCGCTTTCCC
CGCTTTCCCA	GCTTTCCCAA	CTTTCCCAAA	TTTCCCAAAG	TTCCCAAAGA
TCCCAAAGAT	CCCAAAGATA	CCAAAGATAT	CAAAGATATG	AAAGATATGC
AAGATATGCC	AGATATGCCC	GATATGCCCT	ATATGCCCTT	TATGCCCTTC
ATGCCCTTCG	TGCCCTTCGA	GCCCTTCGAG	CCCTTCGAGC	CCTTCGAGCT
CTTCGAGCTG	TTCGAGCTGG	TCGAGCTGGC	CGAGCTGGCA	GAGCTGGCAG
AGCTGGCAGA	GCTGGCAGAA	CTGGCAGAAA	TGGCAGAAAT	GGCAGAAATA
GCAGAAATAC	CAGAAATACC	AGAAATACCT	GAAATACCTG	AAATACCTGC
AATACCTGCG	ATACCTGCGG	TACCTGCGGG	ACCTGCGGGT	CCTGCGGGTA

(read across and down - 1 to 1000)

Table 4: First 1000 Observation of DNA Sequence

References

[1] Alan Agresti. *An Introduction to Categorical Data Analysis*. Wiley, 2007.

[2] Alex Bateman Benjamin Schuster-Böckler. An introduction to hidden markov models. https://currentprotocols.onlinelibrary.wiley.com/doi/full/10.1002/0471250953.bia03as18. Accessed: 01.06.2007.

[3] Pat Brans. Markov models. https://www.techtarget.com/whatis/definition/Markov-model. Accessed: August, 2022.

[4] Rongbo Chen, Haojun Sun, Lifei Chen, Jianfei Zhang, and Shengrui Wang. Dynamic order markov model for categorical sequence clustering. *Journal of Big Data*, 8(1):1–25, 2021.

[5] DeepAI. Hidden markov model. https://deepai.org/machine-learning-glossary-and-terms/hidden-markov-model.

[6] DeepAI. Markov model. https://deepai.org/machine-learning-glossary-and-terms/markov-chain.

[7] Konstantinos Fokianos and Benjamin Kedem. Regression theory for categorical time series. *Statistical science*, 18(3):357–376, 2003.

[8] Clay Ford. Getting started with multinomial logit models. https://data.library.virginia.edu/getting-started-with-multinomial-logit-models/.

[9] Clay Ford. National center for biotechnology information. https://www.ncbi.nlm.nih.gov/assembly/GCF_001293415.1/.

[10] Jim Frost. Time series analysis intoduction. https://statisticsbyjim.com/time-series/time-series-analysis-introduction/. Accessed: August, 2020.

[11] Jeremy Jackson Adrian Hernandez Christopher Williams Calvin Lin Jimin Khim Henry Maltby, Worranat Pakornrat. Markov chains. https://brilliant.org/wiki/markov-chains/.

[12] Vishwanathan Iyer and Kaushik Roy Chowdhury. Spectral analysis: Time series analysis in frequency domain. *IUP Journal of Applied Economics*, 8, 2009.

[13] Patricia A Jacobs and Peter AW Lewis. Stationary discrete autoregressive-moving average time series generated by mixtures. *Journal of Time Series Analysis*, 4(1):19–36, 1983.

[14] Heinz Kaufmann. Regression models for nonstationary categorical time series: asymptotic estimation theory. *The Annals of Statistics*, pages 79–98, 1987.

[15] Gebhard Kirchgässner, Jürgen Wolters, and Uwe Hassler. *Introduction to modern time series analysis*. Springer Science & Business Media, 2012.

[16] NIST. Definitions, applications and techniques of time series analysis. https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc41.htm.

[17] NIST. Introduction to time series analysis. https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm.

[18] NobelPrize.org. Daniel l. mcfadden – biographical. https://www.nobelprize.org/prizes/economic-sciences/2000/mcfadden/lecture/.

[19] Randy Louis Ribler. *Visualizing categorical time series data with applications to computer and communications network traces*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[20] Ian Stewart. *Seventeen equations that changed the world*. Profile Books, 2012.

[21] David S Stoffer, David E Tyler, and Andrew J McDougall. Spectral analysis for categorical time series: Scaling and the spectral envelope. *Biometrika*, 80(3):611–622, 1993.

[22] David S. Stoffer, David E. Tyler, and David A. Wendt. The spectral envelope and its applications. *Statistical Science*, 15(3):224–253, 2000.

[23] Christian H Weiß. *An introduction to discrete-valued time series*. John Wiley & Sons, 2018.

[24] Christian H Weiß and Rainer Göb. Measuring serial dependence in categorical time series. *AStA Advances in Statistical Analysis*, 92(1):71–89, 2008.

[25] wikipedia. David cox (statistician). https://en.wikipedia.org/wiki/David_Cox_(statistician).

[26] Xiaoa Zhen. *Categorical Time Series*. PhD thesis, University of Georgia, 2008.