Estimating Lower Bounds for Time Series Prediction Error

by

Saeyoung Rho

B.S. KAIST (2014)

M.S. KAIST (2016)

Submitted to the Institute of Data, Systems, and Society
in partial fulfillment of the requirements for the degrees of
Master of Science in Technology and Policy
and
Master of Science in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2020

Author .....
Institute of Data, Systems, and Society
August 3, 2020

Certified by .....
Devavrat Shah
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Noelle Eckley Selin
Associate Professor, Institute for Data, Systems, and Society and
Department of Earth, Atmospheric and Planetary Sciences Director,
Technology and Policy Program

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Estimating Lower Bounds for Time Series Prediction Error

by

Saeyoung Rho

Submitted to the Institute of Data, Systems, and Society
on August 3, 2020, in partial fulfillment of the
requirements for the degrees of
Master of Science in Technology and Policy
and
Master of Science in Electrical Engineering and Computer Science

Abstract

Research on how to evaluate the time series prediction algorithms are relatively underinvestigated compared to those to develop prediction algorithms. This research presents a way to estimate lower bounds for a time series prediction error by utilizing the conditional entropy rate, which allows us to take the inherent difficulty of a problem into account. The main focus of this research is on a discrete time series composed of discrete random variables, and stationarity of the time series is assumed. In this thesis, the lower bound is estimated based on the Fano's inequality, which shows the relationship between the conditional entropy rate and prediction error. Therefore, a method to approximate the entropy rate using the Lempel-Ziv compressor is suggested as a subroutine. Also, a discretization method is introduced to adopt this approach to real-valued sequences. Finally, the method is validated for both discrete and continuous distributions, and applications with real-world datasets are demonstrated. The proposed error lower bound can serve as an objective criterion to evaluate the current status of the algorithm and has the potential to aid the technocratic knowledge assessment process in science that involves discrete time series prediction problem.

Thesis Supervisor: Devavrat Shah
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

It takes a village to raise a child, and I needed the whole universe to finish this thesis. I want to thank everyone who directly or indirectly helped me during the past few years, especially my research group's and Professor Shah's endless support. The interactions I had with them were an essential key to complete my journey at MIT. It was though to challenge myself to the next level, but I learned what unconditional love and true friendship mean.

My first and deepest gratitude goes to my best friend, mentor, soul mate, and mom. She has taught me the power of love, infinite patience, and how to overcome the pain. I would also like to thank my academic mother, Dr. Lim, who has shown me how to be an attentive listener, a sincere supporter, and a great leader. Without these two amazing women's sacrifice and guidance, I wouldn't have been myself today.

I would also like to mention an unbelievably strong sisterhood I was able to share with awesome people around me. One has convinced me to stop playing with the imposter syndrome and choose to come to MIT. One had offered me a hug when I was absolutely alone and continually suspecting my ability. One has shown me how to give first before wanting to have one. One has taught me the beauty of mathematics and saved me from the swamp of Russian men. One has proven that art and creativity make our lives delightful. One has enlightened me with ever-evolving ideas to construe the world. One has infused me with a hope that the world may be able to change to a better place. And, one has not just stood up for me but ridden on the roller coaster together throughout this adventure.

Being a board member of the Graduate Women at MIT has also contributed to my identity as a winner of the "Most Likely to Bring Out Your Badass Feminist" award. I have met so many inspiring and unapologetic young women who were not only rocking in science but also writing the new history for the next generation of girls.

I want to remember warm hearts and kind words from all the shiny people around me. I will try to do so as hard as I did to memorize the Gaussian distribution's density function, which I still sometimes forget but eventually remember.

학위를 마치는 데 도움 주신 모든 분들께 감사의 말씀 드립니다.
여기까지 오는 길을 가장 가까이에서 지켜보고 응원해주신 어머니와,
이 학위가 완성되는 것을 보지 못하고 떠나신 할머니에게 이 논문을 바칩니다.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Time series is a chronologically observed sequence of data, and time series modeling is a popular domain that became one of the core elements structuring our modern society. As the sensing and storing technologies advanced in the past decades, so many machines are now recording our daily lives with timestamps: social media usage, online purchase history, and electronic health records. The availability of time series datasets propelled research on time series modeling so as to better understand the phenomena and also to better predict the future. Even when you are reading this, some researchers are developing yet another algorithm on top of the hundreds of preexisting ones.

The time series modeling algorithms are not only for the folks on the Wall Street, but also utilized in numerous steps of the policymaking processes. For example, if we can model the patient's arrival to the emergency room, we could allocate resources appropriately to reduce wait times. If we can predict the road traffic, we could even dynamically change the public transit fare to attract more people. In all cases, it is true that the quality of prediction creates a huge impact on the society and people. However, there is always a possibility that the predictions are inaccurate, and different group of people will be affected by different outcomes. Still, there is a clear benefit of using those predictions for the betterment of society.

This gives rise to the need for an objective criterion that takes the *inherent difficulty*

of a problem into account to assess the performance of time series prediction algorithms. In most of the current papers, researchers compare the algorithm's performance to another algorithm's. This allows us to choose the current best practice, however, if all of those are performing arbitrarily bad, one might say that we should just not rely on those predictions. If we could incorporate the difficulty of the problem into the assessement metric, we would have more in-depth discussions whether or not this technology is mature enough to be utilized.

Ultimately, the goal of this thesis is to propose a benchmark to evaluate the performance of time series prediction algorithms. In a conceptual level, this could be an time series counterpart of R^2 score in regression problems. The R^2 score compares how well this model explains the dependent variable compared to the naive average of all datapoints. What this thesis will propose is the lower bound of prediction error, an evaluation metric that takes into account the *inherent difficulty* of a given time series prediction problem.

1.2 Overview

With the goal of proposing an objective evaluation metric in mind, this thesis adopts the idea of entropy, a measure of uncertainty, to capture the temporal correlation of the time series sequence. For a time series prediction, one problem can be fundamentally easier to predict than the other due to its underlying distribution, i.e., if a source outputs the same value every time, it will be very easy to approach almost 0 error rate after few observations. By obtaining the lower bound of error via entropy approximation, we can include the fundamental difficulty of a problem in the assessment.

This thesis will propose a two-step method based on a theoretical foundation to derive lower bounds for the probability of error for a given time series prediction problem: 1) to estimate the entropy rate; and 2) to obtain the lower bound via Fano's inequality. The entropy rate of a sequence can be approximated by a compression ratio, which turns out to be the same as the conditional entropy given that the sequence is stationary. Fano's inequality shows the relationship between the classification error

and the conditional entropy. Using these two main theorems, we can obtain the lower bound for a classification error. One caveat is that the computed lower bound is an approximation, not a theoretical value.

The suggested error lower bound allows us to take the inherent difficulty of a problem into account when assessing the performance of a prediction algorithm. This approach is best suited for categorical time serieses (e.g., a sequence of human behaviors), but can be also applied for time serises with finitely countable alphabets (e.g., binary, ternary seqeunces). A real-valued sequence should be discretized to be able to go through the same procedure to obtain the error lower bound. I believe that this error lower bound could serve as a standard to objectively evaluate the current status of the algorithm, so that we can prevent endeavours to an impossible problem as the performance approaches to the (approximated) theoretical error lower bound.

The contribution of this paper can be summarized into three parts. First, an algorithm was proposed to approximate the entropy rate of a time series with discrete values. Second, a method was presented to estimate the error lower bounds for a given time series prediction problem. Lastly, the role of this error lower bound estimation technique in science is discussed. The suggested error lower bound can serve as a standard to objectively evaluate the current status of the algorithm for a particular problem. As it incorporates the inherent randomness of the time series, it can prevent endeavors to solve an impossible problem as the performance approaches to the theoretical optimal. Future research could focus on how to generalize this concept to remove the stationarity assumption.

The overall structure of this thesis is as follows. Chapter 2 explains the theoretical foundation and argues the legitimacy of the proposed error lower bound. Chapter 3 suggests a two-step method to estimate a prediction error lower bound of a time-series sequence based on Fano's Inequality, and validates the method for both discrete and continuous random variables. The validation for continuous random variables involves the discretization method and the Kalman filter estimates. Chapter 4 demonstrates the use of this method with some real-world datasets. Chapter 7 highlights the main contributions of this work to both machine learning community and technology policy

community. Finally, Chapter 5 outlines the limitations and future directions of this research and comments on the use of the proposed error lower bound as a technocratic knowledge assessment tool.

1.3 Problem Statement

A discrete time series is a chronologically observed sequence of data, which is one of the most common forms of data available across domains. Predicting the next value has become a central question in scientific research, and hundreds of algorithms for time series prediction have been proposed as the availability of time series datasets increased. While there is a solid amount of work to propose time series prediction algorithms, we lack a tool to assess the performance of those algorithms. For example, R^2 score is one way to assess the goodness of fit of a regression model—it compares how well this model explains the dependent variable compared to the naive average of all data points. A similar concept can be also applied to time series modeling problems, and that is the main goal of this research.

For a time series prediction, one problem can be fundamentally easier to predict than the other due to its underlying distribution, i.e., if a source outputs the same value every time, the error rate can easily approach zero after few observations. To take into account the temporal correlation of the time series, this thesis adopts the idea of conditional entropy rate, a measure of uncertainty of a sequence. The entropy rate of a stochastic process can be approximated using a compression ratio, which turns out to be the same as the conditional entropy rate when the sequence is stationary. Once we obtain the conditional entropy, Fano's inequality enables us to obtain the lower bound for a classification error.

As the proposed method is based on Fano's inequality, this thesis will mainly focus on a time series with a finite alphabet size. Since the probability of error equally penalties all the wrong predictions—i.e, does not capture the regression error—, nominal data type is best suited for this evaluation metric. Nonetheless, it can be applied to any discrete-valued sequences. For a real-valued sequence, we apply a

discretization method to pre-process the data and make it discrete.

To sum up, in this thesis, we focus on a discrete-valued discrete-time series that can be described as follows: a time series $\{X_t\}$ with time index $t \in \mathbb{N}$ and $X_t \in \mathcal{X}$ with alphabet size $|\mathcal{X}| = n$. When the set $\mathcal{X}$ is infinite, we discretize the sequence into a number of bins and treat it as a discrete-valued sequence.

1.4 Related Work

Time series analyses is a well-studied domain with a rich literature. For textbook style references, which include traditional methods such as ARIMA, refer to [7, 6, 15, 22]. Additionally, there are connections to the theory of stochastic processes and information theory (cf. [9, 25, 21, 13]), and latent structures a la Hidden Markov Models (HMMs) (cf. [20, 5] Recently, there has been considerable interest in matrix based methods [2, 28, 29] and neural network methods [8, 23, 24].

One of the main objectives to fit a model is to predict future values, and researchers have developed metrics to evaluate the performance of the trained model such as mean absolute scaled error (MASE) [17] and symmetric mean absolute percentage error (SMAPE) [18]. In this paper, we will focus on simple error metircs, a classification error for discrete distributions and a mean-squared error for continuous distributions.

Some researchers took a step further to propose lower bounds for those error metrics through theoretical lenses. Tichavský and Nehorai came up with an idea to obtain a mean-squared error lower bound for discrete time nonlinear filtering problem based on Cramér-Rao bounds [26]. Erdogmus and Principe suggested a tighter bound using a modified version of Fano's inequality [12]. They obtained a lower bound for a classification error using Renyi's information, instead of Shannon's. Later in 2013, minimax risk lower bounds were proposed for a distributed statistical estimation under communication constraints [30]. Similarly, we adopt an information-theoretic framwork, Shannon's definition of entropy and Fano's inqeuality, to suggest lower bounds for a prediction error in time-series scenarios.

As most of these approaches are based on information-theoretic bounds, such as

Fano's inequality, a common subroutine needed in the derivation of a lower bound includes an estimation of (conditional) entropy rate. Using a universial compressor is one of the traditional ways to estimate entropy, as the normalized codelength of any universal code is a consistent estimator of the entropy rate in an asymptotic regime. For instance, Amigó et al. utilized Lempel-Ziv compression to estimate entropy of a binary string [3]. Han et al. focused on estimating entropy rate of a stationary reversible Markov process [16]. Our approach expands these efforts to build an easily implementable algorithm to estimate the entropy rate of a stiationary process, by building a regression model between the compression ratio and theoretical entropy.

Chapter 2

Background

2.1 Time Series

This thesis focuses on discrete time series with a finite alphabet size. In this section, we will review some basics around the time series and its analysis.

2.1.1 Continuity of the Time Index

Time series can be divided into several sub-groups, and one way to categorize them is whether the time index is continuous or discrete—i.e.,a continuous time series and a discrete time series. In the real world, most signals will be analog, meaning that a signal will have a continuous time index. One can think of a function of time $f(t)$ that outpus some number at any given $t \in \mathbb{R}$. This setup has a continuous domain for the time t, and most of the time series you observe in the real world will have it in this format. For example, the movement of an object will be a continuous time series (the trace of location indexed by continuous time).

Then, imagine what happens if we log the observation of this continuous signal in a digital storage. Once we measure the location and record it with time, the signal automatically becomes a discrete-time data. Therefore, most of time serieses available for analysis are a sequence of data taken at successive equally spaced time points. This is similar to a sequence indexed by natural numbers, instead of a function on a

real line. In this thesis, we will only focus on discrete-time series.

2.1.2 Type of Random Variables

Another way to categorize time series is to look at the type of data measured at each timepoint. We can also divide them if the random variables composing the time series are continuous random variables (e.g., $X_t \in [0, 1]$) or discrete random variables (e.g., $X_t \in \{1, 2, 3, ...\}$). Discrete random variables can have categorical values, instead of numbers, such as names or likert scale labels. When the data are nominal ("banana", "apple", or "orange") or ordinal ("very good", "good", or "bad"), we call it a categorical time series. Note that the ordinal values have a relative relationship to each other that can be ordered and matched to some numerical system. The alphabet size is the cardinality of the sample space and is empirically finite for most cases in the real world.

The main focus of this thesis is a discrete-valued time series, because we build our theory upon the Fano's Inequality. Since the probability of error term that appears in the Fano's Inequality is defines as $P(e) = \mathbf{P}(X_t = x, \hat{X}_t \neq x)$, the best scenario to apply this method is for the nominal data or a binary classification. In other words, this definition of probability of error cannot capture the distance, or a regression error, and penalize all incorrect answers with the same weighting.

2.2 Entropy Estimation via Lempel-Ziv

2.2.1 Entropy Rate and Conditional Entropy Rate

To begin with, we define *entropy* and *conditional entropy* of a discrete random variable as following. Let $X \in \mathcal{X}$ be a discrete random variable with probability mass function $p(x)$.

Definition 1 *The entropy of X is defined by*

$$h(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be two discrete random variables with joint probability mass function $p(x, y)$.

Definition 2 *The conditional entropy of $Y|X$ is defined by*

$$h(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(ylx),$$

where $p(y|x)$ is a marginal distribution of y given x.

We define similar concepts not just with one random variable, but with stochastic processes. Let $\{X_t\} = X_1, X_2, ..., X_T$ be a discrete stationary stochastic process where $X_t \in \mathcal{X}$, $\forall t = 1, 2, ..., T$ and $|\mathcal{X}| = n$, $n \in \mathbb{N}$.

Definition 3 *The entropy rate of a stochastic process $\{X_t\}$ is defined by*

$$H(\mathcal{X}) = \lim_{T \to \infty} \frac{1}{T} h(X_1, X_2, ..., X_T)$$

when the limit exists.

Definition 4 *The conditional entropy rate of a stochastic process $\{X_t\}$ is defined by*

$$H'(\mathcal{X}) = \lim_{T \to \infty} h(X_T|X_{T-1}, X_{T-2}, ..., X_1)$$

when the limit exists.

2.2.2 Lempel-Ziv Algorithm

The Lempel-Ziv compressor is a universal lossless data compression algorithm that will be utilized throughout this paper [31]. In this section, we will review how Lempel-Ziv compressor works by reviewing some of the materials in Chapter 5 of Data Compression [19]. Let the datavector $X = \{X_1, X_2, ..., X_T\}$, a time series data that we want to analyze. Lempel-Ziv algorithm parses the datavector according to a certain rule. The first block is always $B_1 = X_1$ and B_1 is added to the dictionary. Then, the next block

in the parsing is the shortest prefix of $\{X_2, ..., X_T\}$ that is not in the dictionary yet (that is, not equal to X_1). If the second parsing was $B_2 = X_2, ..., X_j$, B_2 is added to the dictionary and the next round is to find the shortest prefixt of $\{X_{j+1}, ..., X_T\}$.

The final block composition $B = \{B_1, ..., B_t\}$ is then represented as a pair of integers. The block of length 1, such as the first block, is represented as $(0, B_i)$. If the length is greater than one, it is represented as (i, s), where s is the last symbol of B_j and i is the index of the block in the dictionary that is the same with the block B_j without the last digit (s). Note that the length of each block B_i are not fixed and they are unique except for the last one B_t by construction.

Finally, each pair (i, s) is replaced by an integer $ki + s$, where k is the size of the alphabet (the cardinarlity of the sample space). The sequence is now composed of integers $I_1, ..., I_t$. The last step is to conver this to binary numbers and pad zeros on the left so that the overall length of the string of bits assigned to I_j is $\lceil \log_2(kj) \rceil$. The concatenation of those integers expressed in binary with zero paddings is the final encoding of the LZW compression.

For example, imagine that X is a binary sequence (composed of only zero's and one's). It starts with a dictionary of the basic set of alphabets (e.g., $\{\mathbf{0} : 0, \mathbf{1} : 1\}$ for a Bernoulli process) and adds a previously unobserved pattern to its dictionary as it reads the sequence in. The **Lempel-Ziv Algorithm** can be summarized as follows:

1. Initialize the dictionary to contain all blocks of length one $\{\mathbf{0} : 0, \mathbf{1} : 1\}$.
2. Search for the longest block **W** which has appeared in the dictionary.
3. Encode **W** by its index in the dictionary.
4. Add **W** followed by the first symbol of the next block to the dictionary.
5. Go to Step 2.

2.2.3 Entropy Rate Estimation via Lempel-Ziv Compression

From now on, we will assume that the sequence of interest is *stationary*.

Definition 5 *A stochastic process* $\{X_t\}$ *is called stationary if*

$$F_X(x_1, ..., x_T) = F_X(x_{1+\tau}, ..., x_{T+\tau})$$

for all τ *and time index* $1, ..., T$ *and for all* $T \in \mathbb{N}$*, where* $F_X(x_{1+\tau}, ..., x_{T+\tau})$ *is the cumulative distribution function of the unconditional joint distribution at times* $1 + \tau, ..., T + \tau$*.*

In other words, the statistical property at some time index remains unchanged as the time indeices are shifted.

Now, imagine we compress a stationary sequence $X = \{X_1, ..., X_T\}$. Using the Lempel-Ziv encoder, we will compress the input sequence X. Following the steps in section 2.2.2, we can define the parsing B and the encoding W.

- Original sequence $X = \{X_1, ..., X_T\}$
- Lempel-Ziv parsing $B = \{B_1, ..., B_{\tilde{T}}\}$
- Lempel-Ziv encoding $W = \{W_1, ..., W_{\tilde{T}}\}$

For simplicity, let us assume that the sequence X is binary, i.e., the original sequence length is T. Traditionally, the compression ratio is defined as

$$\tilde{R} = \frac{\sum len(W_i)}{T},$$

where $len(\cdot)$ measures the length of the sequence. This ratio, the length of the compressed sequence divided by the length of the original sequence, will approximate the expected cordword length per symbol.

In other words, $\tilde{R}$ is the codeword length per symbol. We also know that the Lempel-Ziv is asymptotically optimal, meaning that $\tilde{R}$ will tend to the minimum expected codeword length per symbol, $\tilde{R}^*$. Then, we know the following is true:

Theorem 1 *Let the minimum expected codeword length per symbol* $\tilde{R}^*$*. Then, for a*

stationary stochastic process X,

$$\tilde{R}^* \to H(\mathcal{X}),$$

and the proof can be found in Cover's textbook [10].

In this thesis, we will define the compression ratio using the number of parsed bins ($\tilde{T}$), instead of the actual encoded sequence length ($\sum len(W_i)$).

Definition 6 *The compression ratio (R) for a sequence $X_1, ..., X_T$ of length T using the Lempel-Ziv compressor is defined as*

$$R = \frac{\tilde{T}}{T},$$

where $\tilde{T}$ is the number of parsed bins using Lempel-Ziv algorithm.

We can rewrite R as

$$R = \frac{\tilde{T}}{T} = \frac{\sum len(W_i)}{T} \cdot \frac{\tilde{T}}{\sum len(W_i)} = \tilde{R} \cdot \frac{\tilde{T}}{\sum len(W_i)}.$$

Remember that the length of W_i's are fixed as $\lceil \log_2(ki) \rceil$ and i ranges from 1 to $\tilde{T}$. Therefore, for a fixed length T, the value $C := \frac{\tilde{T}}{\sum len(W_i)}$ remains approximately constant. This gives a base for why there exists a linear relationship between the compression ratio R and the entropy $H(\mathcal{X})$.

We can go further to investigate how the linear relationship holds. We know that a universal code achieves average codeword length per symbol that is at most a constant times the optimal possible for that source [19]. Specifically for the Lempel-Ziv case, the following hods.

Theorem 2 *For a Lempel-Ziv encoding W_i's of $X = \{X_1, ..., X_T\}$,*

$$\sum len(W_i) \leq T \cdot H(\mathcal{X}) + T \cdot \epsilon_T,$$

where ϵ_T only depends on the sequence length T.

Rearranging this inequality, we get

$$\begin{aligned}\sum len(W_i) &\leq T \cdot H(\mathcal{X}) + T \cdot \epsilon_T \\ \frac{\sum len(W_i)}{T} &\leq H(\mathcal{X}) + \epsilon_T \\ \frac{\tilde{T}}{T} \cdot \frac{\sum len(W_i)}{\tilde{T}} &\leq H(\mathcal{X}) + \epsilon_T.\end{aligned}$$

By defining $C := \frac{\tilde{T}}{\sum len(W_i)}$, we get

$$R \leq C(H(\mathcal{X}) + \epsilon_T).$$

The value C is not a constant, but is empirically constant for a fixed T. Therefore, at the optimal compression power, R is in a linear relationship with $H(\mathcal{X})$ with an intercept term.

Finally, we can exchange the entropy rate with the conditional entropy rate as they are equal to each other [10].

Theorem 3 *For a stationary stochastic process, both $H(\mathcal{X})$ and $H'(\mathcal{X})$ exist and are equal*

$$H(\mathcal{X}) = H'(\mathcal{X}).$$

2.2.4 Discretizing Continuous Distributions

All of the above statements are about a sequence generated from sources with a discrete distribution with a finite alphabet size. For a real-valued sequences, we will discretize the sequence into 2^k bins and approximate the original distribution.

Let $\{Y_t\} = Y_1, Y_2, ..., Y_T$ be a real-valued stationary stochastic process where $Y_t \in [0, 1]$, $\forall t = 1, 2, ..., T$. Let the discretizing function

$$b_k(Y_t) = i \text{ if } Y_t \in B_{k,i}$$

where $B_{k,i} = \{y | \frac{i-1}{2^k} \leq y \leq \frac{i}{2^k}\}$, for $i = 1, ..., 2^k$. Using the discretized $X_t = b_k(Y_t)$, we can apply the same logic to the continuous random variables.

When the support of the random variable is not $[0, 1]$, we can transfer the observation to range in $[0, 1]$ by subtracting and scaling the values.

$$\tilde{Y_t} = \frac{Y_t - Y_{(1)}}{Y_{(T)} - Y_{(1)}},$$

where $Y_{(1)} < Y_{(2)} ... < Y_{(T)}$ are order statistics.

2.3 Fano's Inequality

Finally, we can apply Fano's Inequality to obtain the prediction error lower bound $P(e) = \mathbf{P}(X_t = x, \hat{X_t} \neq x)$, where $\hat{X_t} = f(X_1, ..., X_{t-1})$ is your prediction for X_t.

Theorem 4 *(Fano's Inequality) Let $\hat{X_T}$ be a function of $X_1, ..., X_{T-1}$ and $h_2(p)$ be a binary entropy function.*

$$H(X_T|X_{T-1}, ..., X_1) \leq h_2(\epsilon) + \epsilon \log(|\mathcal{X}| - 1)$$

where $\epsilon = \mathbf{P}(X_T \neq \hat{X_T})$ [10].

Note that in a discretized continuous distribution case,

$$\begin{aligned} \epsilon &= \mathbf{P}(X_T \neq \hat{X_T}) \\ &= \mathbf{P}(b_k(Y_T) \neq \hat{X_T}) \\ &= \mathbf{P}(Y_T \notin B_{k,\hat{X_T}}). \end{aligned}$$

Note that this is not the most useful definition of error in regression setting. However, it is still one objective to achieve and a good enough metric that can tell us an actionable insight. For example, obtaining the value of ϵ for many different k's, we can determine the length of confidence interval $(1/2^k)$ that guarantees a lower bound of certain probability of error $\hat{\epsilon}$.

Chapter 3

A Framework for Error Lower Bound Estimation

3.1 A Two-step Method

In this section, a two-step method is proposed to estimate a prediction error lower bound. First, we approximate the entropy rate of a discrete sequence using Lempel-Ziv compression. Next, we apply Fano's inequality to obtain the error lower bound. At the end of this section points out potential sources of noise in the estimation process.

3.1.1 Entropy Rate Estimation

The length of codes for each parsed bin W_i is a function of i: $f(i) = \lceil \log(ki) \rceil$. Figure 3-1 shows this graph, the length of W_i for varying i when the cardinarlity of the sample space is $k = 3$. Note the the x-axis ranges form 0 to 1 million (1,000,000), whereas the y-axis limit is 25. The function is basically a ceiling of a log function, hence it remains constant for longer and longer when we gradually increase the time index(i). This shows that the length of W_i does not change rapidly, and even slower when the time index(i) is bigger.

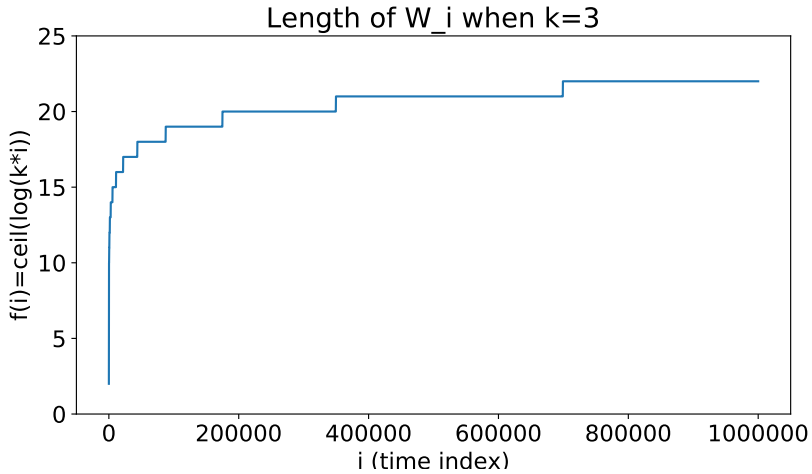

Figure 3-1: The length of W_i for varying i when the cardinarlity of the sample space is $k = 3$; $f(i) = \lceil \log(ki) \rceil$.

Therefore, the length of a compressed sequence is

$$\sum_{i=1}^{\hat{T}} W_i,$$

where $\hat{T}$ is the number of parsed bins using Lepmep-Ziv compressor. This can be approximated by a factor of $\hat{T}$, since W_i does not change rapidly. In other words, the compressed sequence length can be expressed as

$$\beta \cdot \hat{T}. \tag{3.1}$$

Now, recall that the minimum expected codeword length converges to the entropy rate (Theorem 3). This means that if we have a sequence of length T, the compressed seqence length will be

$$T \cdot H(\mathcal{X}) \tag{3.2}$$

as T goes to infinity.

By comparing equations 3.1 and 3.2, we get

$$\beta \cdot \hat{T} \approx T \cdot H(\mathcal{X})$$

$$\beta \frac{\hat{T}}{T} \approx H(\mathcal{X}).$$

Finally, as our definition of the compression ratio R is $\frac{\hat{T}}{T}$,

$$\beta \cdot R \approx H(\mathcal{X}).$$

Therefore, we use a linear model to obtain a relationship between the entropy rate ($H(\mathcal{X})$) and the compression ratio (R).

Based on this, below is a step-by-step explanation on how this linear model can be utilized to estimate the entropy rate. When a sequence $\{X_t\}$ is given, the alphabet size $|\mathcal{X}| = n$ and sequence length T are already decided. We build a regression model between compression ratio R of a random sequence and associated theoretical entropy H by randomly generating s sample sequences and compressing them. When generating sample sequences, we randomly draw a probability vector p, i.e., $\sum_{i=1}^{n} p_i = 1, 0 < p_i < 1$, and generate a sequence of length T from **Multinomial**(p). The linear regression model learns the coeffieicnts β and γ in the form below:

$$H = \beta R + \gamma.$$

Note that the value of γ should be close to 0, given that the proposed method follows our theoretical justification. Once we have a regression model, we compress the given sequence $\{X_t\}$ to measure the compression ratio R^* and use the regression model to approximate the entropy rate

$$H^* = \beta R^* + \gamma.$$

3.1.2 Obtaining Error Lower Bound

Finally, we can find the probability of error ϵ by using Fano's inequality. The right-hand side of Fano's inequality (Theorem 4) is a function of $\epsilon = \mathbb{P}(\hat{X} \neq X)$,

$$f(\epsilon) = h_2(\epsilon) + \epsilon \log(|\mathcal{X}| - 1),$$

and we can approximate the error lower bound

$$\epsilon^* = f^{-1}(H^*). \tag{3.3}$$

As there is no closed-form formulation of $f^{-1}(\cdot)$, we use a gradient descent method to approximate the inverse function within an error bound of 0.001.

3.1.3 Potential Sources of Noise

There are several sources of noise in this approach. The first one is the fundamental stochasticity of the random process generated to buld a regression model. The second one is the error from the inverse function approximation (equation 3.3). Lastly, the approximation made in the equation 3.1 may contribute to inaccurate estimation as well because the range of i may not be in the same flat region of the graph (Figure 3-1).

3.2 Validation of the Method

3.2.1 Linear Model

First, we test if it is indeed appropriate to adopt a linear model. Figure 3-2 shows the compression ratio - entropy linear model fitted using 100 random samples with the same length and alphabet size but varying probability distributions. The probability vector was sampled from a uniform distribution—e.g., each element of $p = [p1, p2, p3, p4, p5]$ was sampled from a uniform distribution over $[0, 1]$ and scaled so that it sums up to 1. The horizontal axis represents the theoretical entropy calculated by the know probability distribution, and the vertical axis is the compression ratio ($\hat{T}/T$). The blue dots correspond to multinomial processed with 5 states, and the orange line is the linear regression fitted with the intercept term, using all the blue dots.

The four plots in Figure 3-2 represent the same experiment for varying sequence length T; $T = 128$ (top left, $R^2 = 0.792$), $T = 256$ (bottom left, $R^2 = 0.906$), $T = 512$

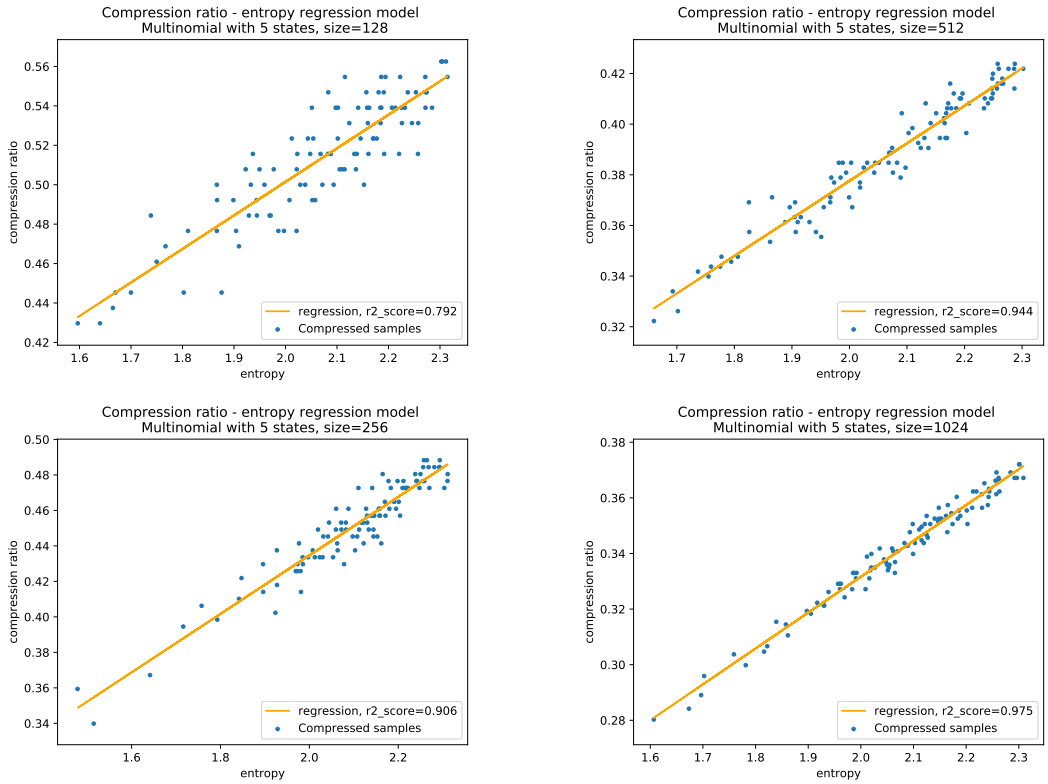

Figure 3-2: The compression ratio - entropy linear model fitting for random sequences with the alphabet size $k = 5$ and sequence length $T = 128$ (top left), $T = 256$ (bottom left), $T = 512$ (top right), $T = 1024$ (bottom right).

(top right, $R^2 = 0.944$), $T = 1024$ (bottom right, $R^2 = 0.975$). The increasing R^2 score reconfirms the observable trend in dots congregating at the regression line. The slope of the regression line decreases as T increseas. This observation meets our expectation that the slope $C := \frac{\tilde{T}}{\sum len(W_i)}$ should decrease as T increases.

Figures 3-3 and 3-4 illustrates how this linear model can be utilized to estimate the unkown entropy of a sequence. In figure 3-3, four different multinomial processes were tested. For example, for the top left plot, the example sequence of length $T = 128$ was generated with a known probability distribution $p = [0.1, 0.1, 0.3, 0.4, 0.1]$. Then, the sequence goes through the Lempel-Ziv algorithm to measure the compression ratio. The red dot on the orange regression line shows the estimated entropy rate via the linear model, as if we do not know its true underlying entropy rate. The red vertical line is the theoretically computed entropy rate of the sequence. The four plots again show the similar result for varying sequence length $T = 128$ (top left), $T = 256$

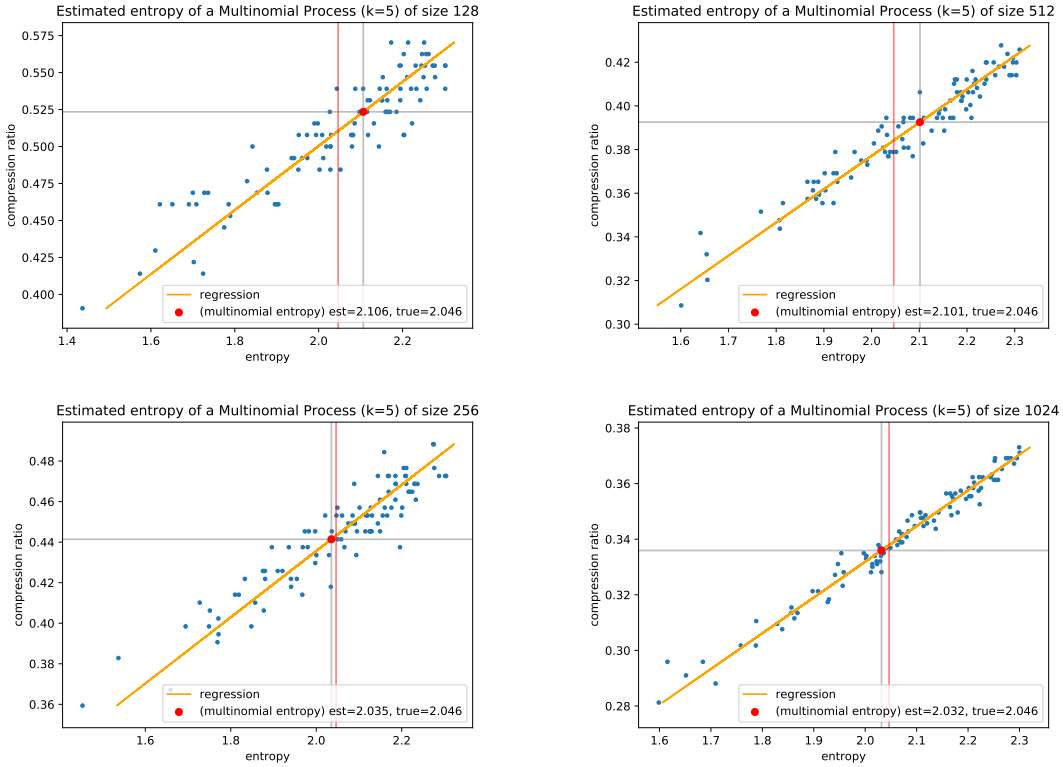

Figure 3-3: Compression ratio - Entropy regression model tested for multinomial process with the alphabet size k=5. The sequence length varies from $T = 128$ (top left), $T = 256$ (bottom left), $T = 512$ (top right), to $T = 1024$ (bottom right).

(bottom left), $T = 512$ (top right), and $T = 1024$ (bottom right).

In Figure 3-4, similar examples are given with Markov processes. The transition matrix P is set to be

$$P = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.3 & 0.2 & 0.3 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.5 & 0.2 & 0.1 \end{pmatrix}.$$

As we have full information about its transition matrix, we can theoretically calculate the true entropy rate of this process, and it is approximately 2.067. Although we know the true value, we will pretend as if we do not know it and try to estimate the entropy rate using our linear model. Similarly, to obtain what is shown on the left top figure,

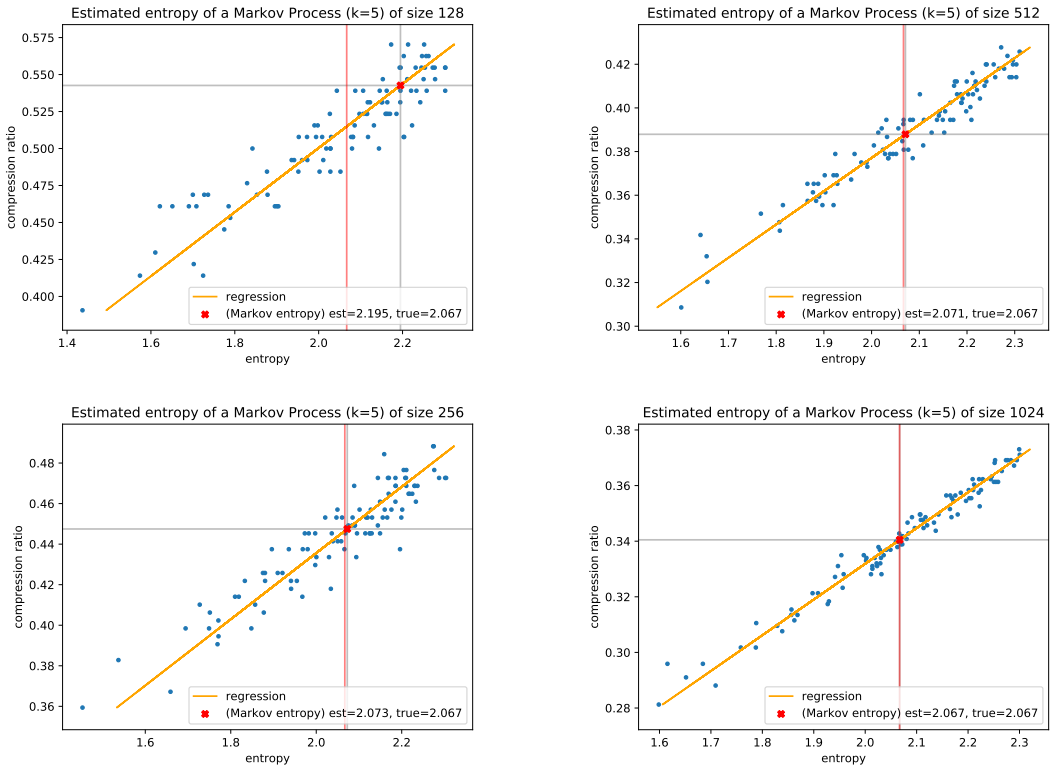

Figure 3-4: Compression ratio - Entropy regression model for varying sequence length T. Tested for Markov process with the alphabet size k=5.

we generate the sequence of length 128 and compress it to measure the compression ratio. The red cross represents the estimated entropy rate, and the red vertical line is the true entropy rate.

Keep in mind that no matter what the underlying distribution of the sequence of interest is, the regression model only depends on the support (the cardinality of the space, k) and the length of the sequence (and of course the number of samples—100 in this case). After constructing the regression model, we obtain the compression ratio of the sequence and infer the entropy rate associated with that compression ratio. The four plots show the data fit into a linear model for all choices of T, with higher accuracy as the length T increases. However, we leave a quantitative analysis on the accuracy of the estimation for the next section.

3.2.2 Entropy Rate Estimation Error

We examined the error of this entropy rate estimation process using i.i.d. sequences (multinomial process, Figure 3-3) and dependent sequences (Markov processes, Figure 3-4). Those plots were just one example to show how it works. In this section, we will run multiple rounds of experiments to observe a more macroscopic trend of the estimation error.

To begin with, let us remind of the two processes, multinomial process and Markov process, and how to calculate the entropy rate from its probability distribution (and a transition matrix).

Multinomial Process Let X_t be independently and identically drawn from **Multinomial**(p), where $\sum_{i=1}^{n} p_i = 1, 0 \leq p_i \leq 1, \forall i = 1, ..., n$. The sample space $\mathcal{X}$ has a cardinarlity of n and the elements of the sequence are independent to each other. The entropy rate of a multinomial process can be calculated using **Definition 1.**

Markov Process Let X_t be a Markov process with a transition matrix P. The elements of this sequence will be dependent to each other, and the entropy rate of a Markov chain can be calculated as

$$h(P) = -\sum_{i,j} \mu_i P_{i,j} \log P_{i,j},$$

where P is its transition matrix and μ is the asymptotic distribution.

To examine how well this estimator approximates the entroy rate, 100 probability vectors (or transition matrices for Markov process) were randomly generated and each produced a sequence of length 1024. Then, 1) the theoretical entropy rate from the probability distribution and 2) the estmated entropy rate using the two-step process were calculated.

Figure 3-5 shows the distribution of discrepancy between the estimated entropy rate and the true entropy rate. The graphs are showing the result of 100 random sequences of length 1024, each from multinomial (top, mean $= -0.003$, standard deviation $= 0.028$) and Markov (bottom, mean $= 0.031$, standard deviation $= 0.029$) processes. The top figure confirms that the estimator is centered around zero, meaning

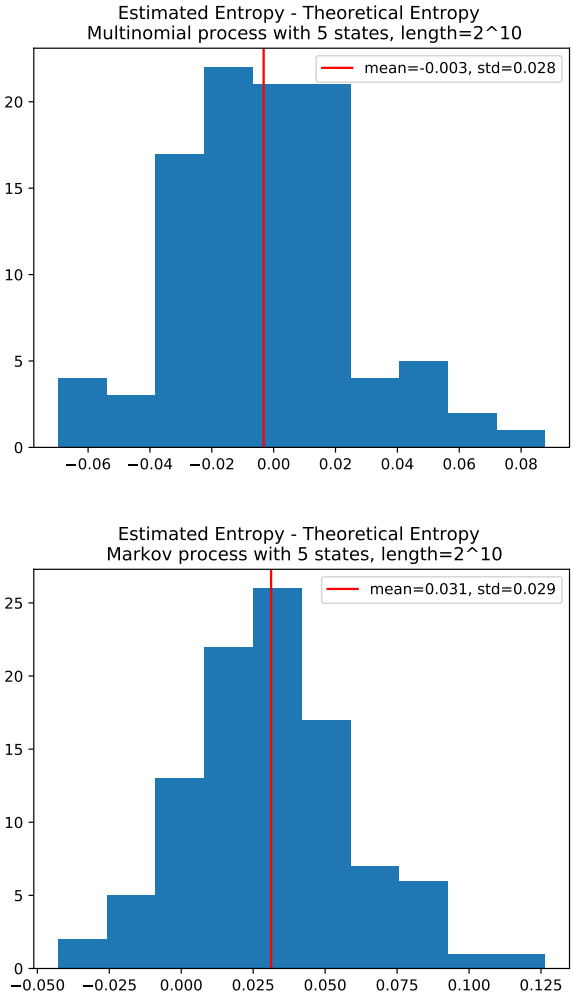

Figure 3-5: The histogram of discrepancy between the estimated entropy rate and the true entropy rate. 100 random samples from multinomial (top) and Markov (bottom) processes.

that this is an unbiased estimator in practice. The bottom figure is slightly off from the zero centerline. There could be several reasons for this phenomenon, but one the major contributors might be the mixing time of the Markov process. If the length ($T = 1024$) is not long enough, the sequence may have not revealed the full behavior of the process. The general tendency of overestimation needs further exploration to explain why.

Figure 3-6 presents a log-log polt of the absolute error and sequence length T, where the same examples were used as in Figure 3-5. The plot on the top refers to the

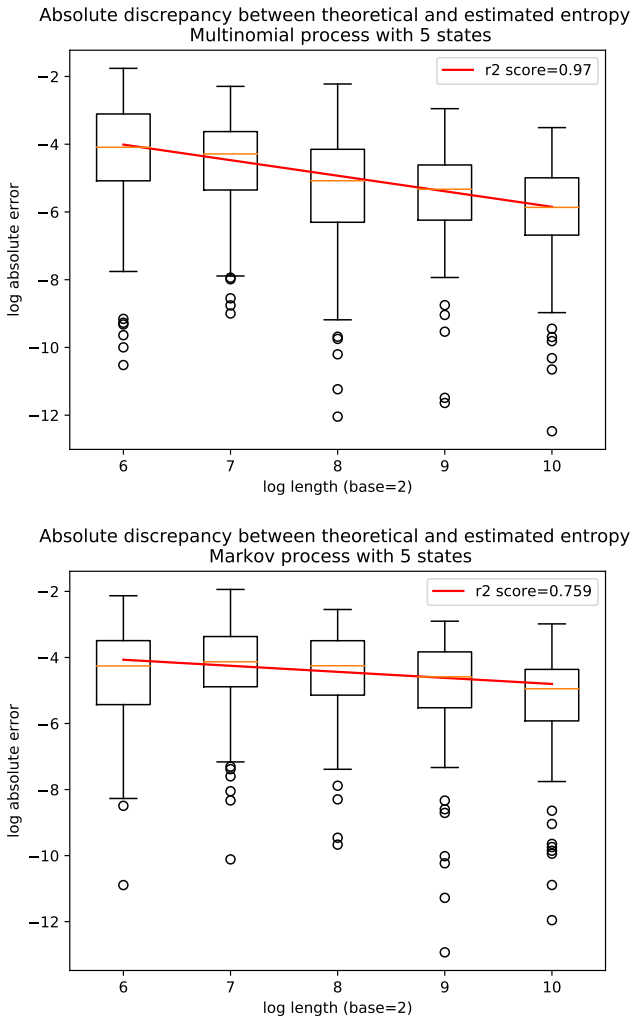

Figure 3-6: A log-log plot of the absolute error and sequence length T. 100 random samples from multinomial (top) and Markov (bottom) processes.

multinomial processes and the one on the bottom refers to the Markov processes. The box whisker plot shows that the error decreases as the sequence length T increases. R^2 score (0.976) close to 1 in the top plot shows that the error decreases in polynomial. In the Markov plot (bottom), the median of error slightly increases from $T = 2^6$ to $T = 2^7$, however, this could be a result of the short sequence length that did not reach the mixing time. For both multinomial and Markov processes, the error decreases as the sequence length (T) increases, and it follows approximately linear descent.

Overall, we can conclude that the proposed two-step method works for both multinomial (independent) and Markov (dependent) processes within 0.03 accuracy

(1 standard deviation), and the accuracy increases in polynomial as the sequence length T increases. The method works better for the multinomial process, however, the accuracy does not differ too much for Markov process and is within the empirical boundary for practical use.

3.2.3 Comparison to Kalman Filter

Kalman filter applied on a Gaussian linear model allows an access to the smallest possible prediction error, as it is an optimal estimator for the underlying data generation process. Hence, we synthetically generated time-series data by Gaussian linear model and compared the prediction error to the lower bound estimation produced by the proposed method.

Let the sequence Y_t be defined by

$$Y_{t+1} = aY_t + Q_t, \tag{3.4}$$

where $Q_t \sim \mathcal{N}(0, 1)$. Similarly as above, we produce $\{X_t\} = \{b_k(Y_t)\}$ with its discretized sample space of size 2^k and apply the method to obtain the error lower bound.

We use the Kalman filter estimation as a tool to assess how good the lower bound is. We produce $Z_t = X_t + R_t$, where $R_t \sim \mathcal{N}(0, 1)$, which will be the observation for the Kalman filter. In this setup, the Kalman filter is the optimal linear filter and thus allow us to use its classification error as a standard to compare with [**?**]. The classification error ϵ for the kalman filter estimates $\hat{Y}_t$ is defined as

$$\epsilon = \mathbf{P}(Y_t \in B_{k,i}, \hat{Y}_t \notin B_{k,i}).$$

Figure 3-7, 3-8, and 3-9 show the error lower bound and Kalman filter classification error for varying k (which defines the number of bins), for different values of a (which denotes the correlation). Note that all tested sequences had length $T = 1024$ and a larger a denotes more correlation between two timestamps (refer to 3.4). The estimated

error lower bound (blue) is below the Kalman filter classification error (orange) for $a = 0.1$ and $a = 0.5$ cases. In $a = 1.0$, the two lines are overlapping, showing the tightness of Fano's bound. We conclude that this does serve as a legitimate error lower bound for continuous distributions as well.

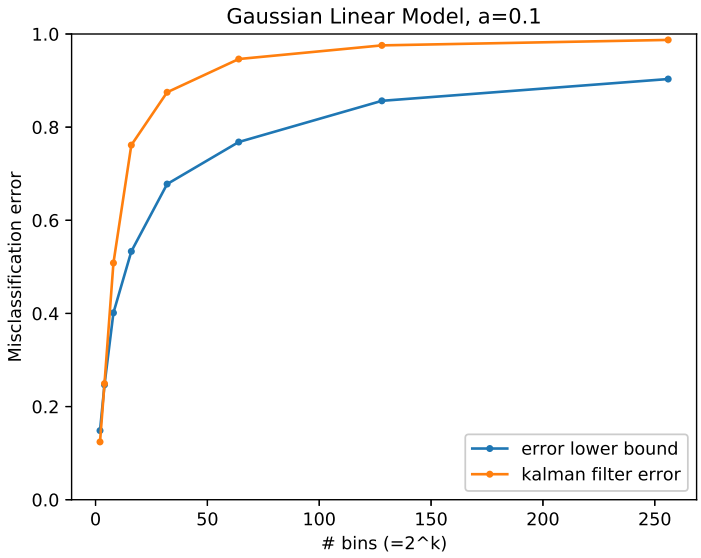

Figure 3-7: The estimated error lower bound (blue) and the Kalman filter classification error (orange) for varying number of bins 2^k. ($a = 0.1$)

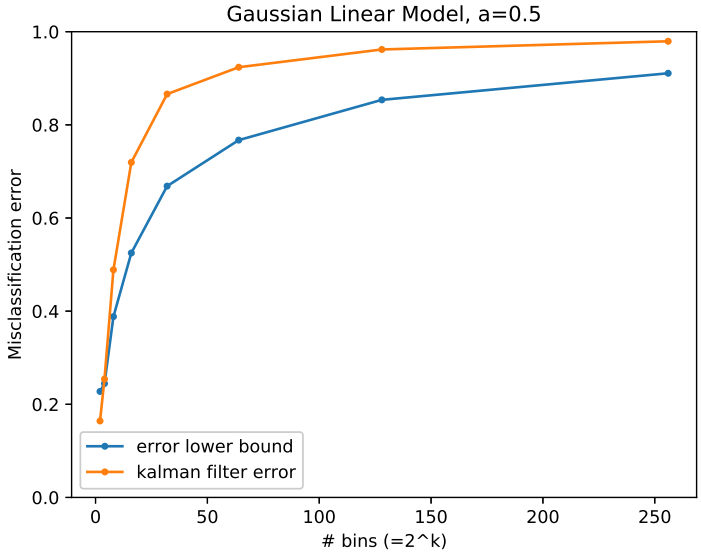

Figure 3-8: The estimated error lower bound (blue) and the Kalman filter classification error (orange) for varying number of bins 2^k. ($a = 0.5$)

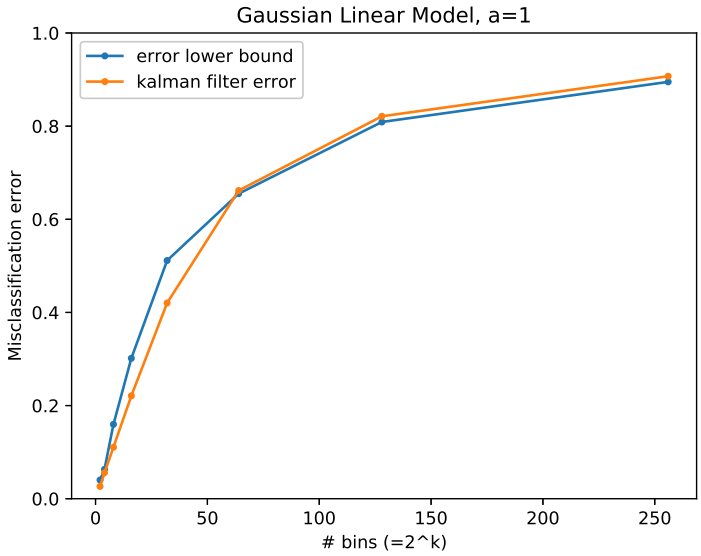

Figure 3-9: The estimated error lower bound (blue) and the Kalman filter classification error (orange) for varying number of bins 2^k. ($a = 1$)

Chapter 4

Demonstration with Real-world Datasets

In this chapter, we assess the performance of the two-step method for error lower bound estimation (section 3.1) and demonstrate how the suggested method can be used with real-world datasets. The example time series covered in this chapter are: sleep stage log data (section 4.1), bitcoin price data (section 4.2), NBA game score data (section 4.3), electricity data (section 4.5), and financial data (section 4.4). Discussion on advantages and disadvantages of applying this method in each case can be found in Chapter 5.

4.1 Sleep Data

Fokianos and Kedem trained models to predict categorical time series sequences [14]. One of them is a sleep data comprised of 4 states: 1) quiet sleep, 2) indeterminate sleep, 3) active sleep, and 4) awake. Figure 4-1 shows this time series with 4 categories logged at 1024 timepoints.

From this sequence of length 1024, we build a regression model using random samples of multinomial with 4 states. Then, we can compress the given sequence and obtain the compression ratio, which will be used to estimate the entropy rate via the regression model (the first step of the method). Finally, we can obtain the prediction

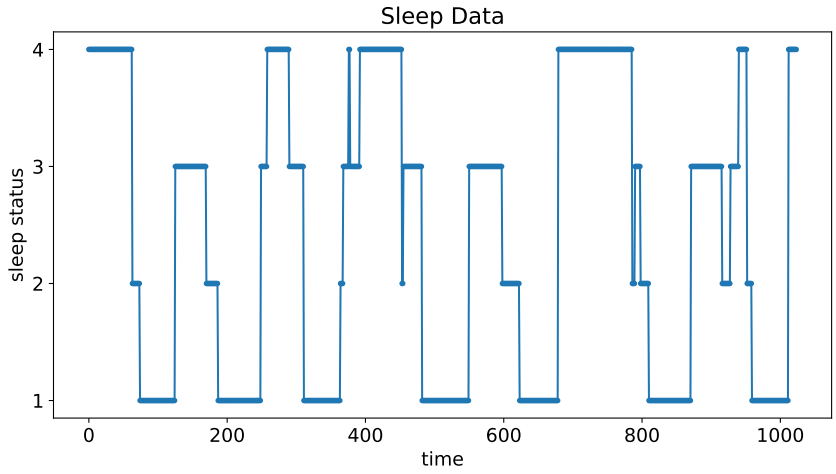

Figure 4-1: sleep data

error lower bound based on the entropy rate estimation. Using the sleep stage data, the two-step method estimated the error lower bound to be 0.0375.

4.2 Bicoin Data

Bitcoin price is one of the most vibrant time series like many stock prices. Although it is unreasonable to condiser that the bitcoin price is stationary, we could assume its stationarity for a relatively short period of time. In Figure 4-2, the Bitcoin price data sampled at 5 seconds interval from 12/1/2014 to 3/31/2015.

tuation vector (x_t) as follows [4].

$$y_t = z_t - z_{t-1},$$

where z_t is the observation at time t, and

$$x_t = \begin{cases} 1 & \text{if } y_t > \theta \\ -1 & \text{if } y_t < -\theta \\ 0 & \text{otherwise.} \end{cases}$$

By setting $\theta = 0$, the fluctuation vector (x_t) will have three values $-1, 0, 1$, each

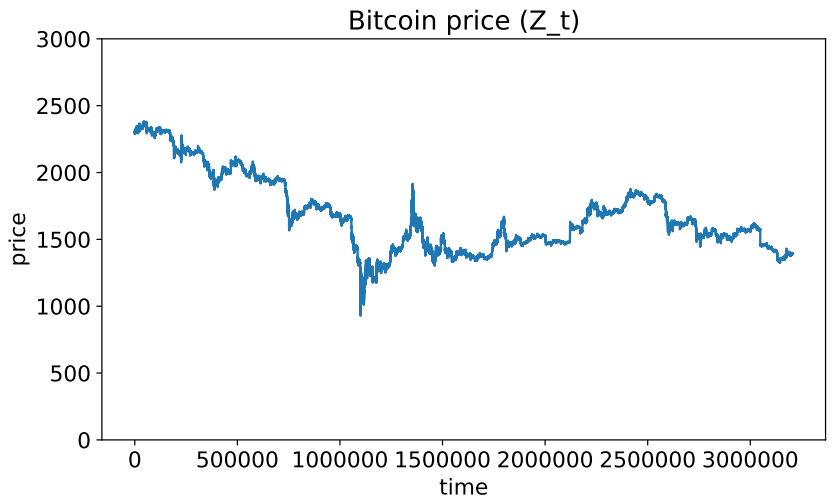

Figure 4-2: Bitcoin price from 12/1/2014 to 3/31/2015 (z_t)

meaning price drop, price stays the same, and price increase, respectively. Figure 4-3 shows the first hundred values of X_t that we obtain from Z_t.

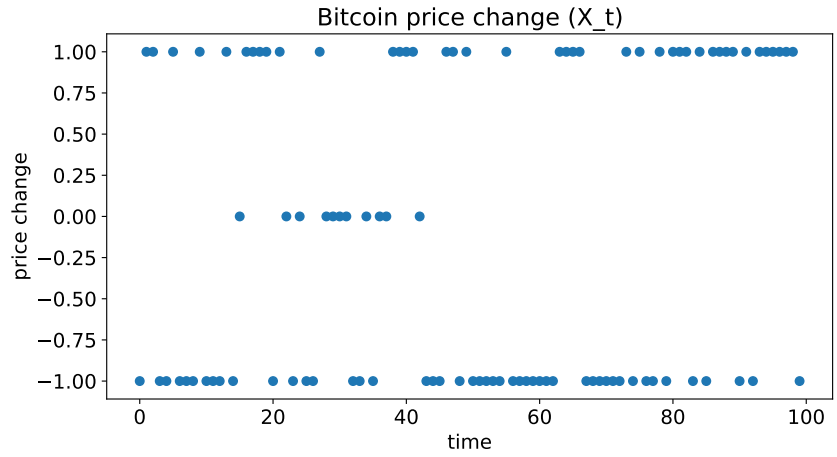

Figure 4-3: Discretized bitcoin price (x_t)

Then, the data for December 2014 was tested to find out the lower bound of prediction error. The sequence X_t was segmented into sub-sequences of length T, and the error lower bound was estimated for each sub-sequence. The test was iterated for varying values of T, and the error lower bound for each equi-length sequence was obtained and plotted in Figure **??**, **??**, and 4-4. The number of tests decreases as T increases, as we tested for a fixed amount of time (1 month). For all choices of T,

$2^{10}, 2^{12}, 2^{14}$, the lower bound for classification error remained around 0.38.

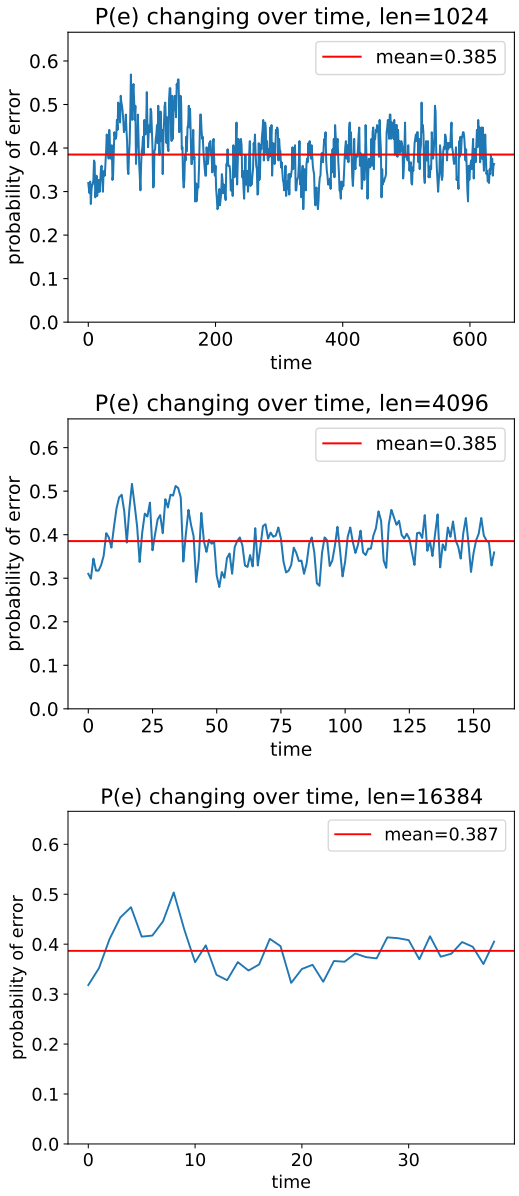

Figure 4-4: The lower bound for classification error obtained for sub-sequences of length $T = 2^{10}$, $T = 2^{12}$, and $T = 2^{14}$

4.3 NBA Data

The National Basketball Association (NBA) game score data is publicly available at the official NBA website (https://stats.nba.com/). Using the play-by-play dataset, the

game score trajectory at 15-second-interval was obtained. The dataset used in this analysis contained 7380 games played between season 2013 and 2018.

For example, a home team's game score trajectory of the first game in season 2013 is plotted on the left side of Figure 4-5. The time scale is in 15 seconds, and the graph is showing 48 minutes of data (from the first quarter to the fourth quarter). From this score trajectory, we can obtain the first order difference, which is shown on the right side of Figure 4-5. The score difference is zero for most of the times, and three points was the maximum score difference made in 15 seconds for this game. In some games, however, six-point-difference was observed albeit it was very rare. Therefore, the number of alphabets (n) was selected differently for each game, based on the observation.

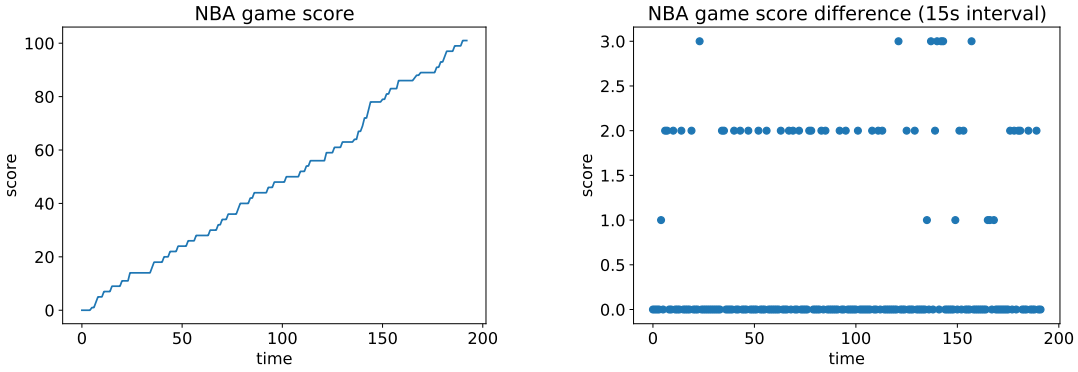

Figure 4-5: NBA score trajectory (left) and the score difference (right)

We can obtain two score trajectories—home and away—from one game, and each score trajectory's prediction error lower bound was estimated via the two-step method. Figure 4-6 shows the distribution of error lower bounds obtained from the games in season 2013 (left) and 2018 (right). Each season comprises of 1230 games, so the total number of trajectories in each histogram is 2460. The mean of the error lower bounds in season 2013 and 2018 were 0.217 and 0.237, respectively. The mean increased by 0.02, which is 2%p. The standard deviation in seadon 2013 and 2018 were 0.025 and 0.024, respectively. The standard deviation remained relatively unchanged compared to its mean.

In Figure 4-7, the box-whisker plot of the error lower bound distribution is shown

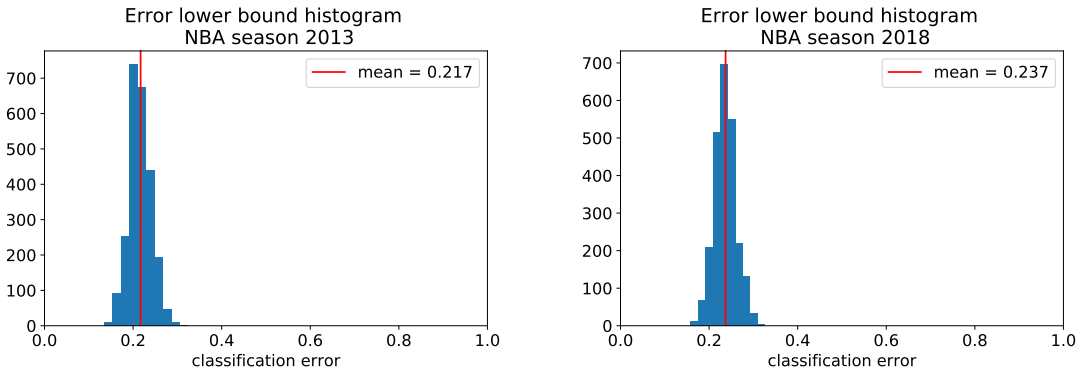

Figure 4-6: Histogram of error lower bounds, season 2013 (left) and 2018 (right)

by each season. From 2013 to 2018, the mean of the distribution tends to increase. With 7380 games in total, 14760 score trajectories are contained in the plot. The distribution seems to be more concentrated around the mean in 2016-2018 than in 2013-2015.

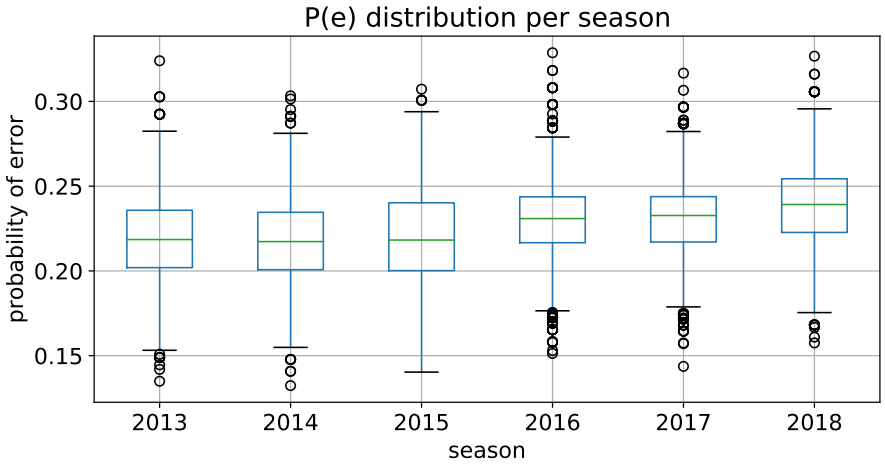

Figure 4-7: Box plot of the error lower bound distribution by season; from 2013 to 2018

4.4 Financial Data

The time series prediction database (tspDB, http://tspdb.mit.edu/) is ad database sepcifically designed for time series that enables predictive query functionality in PostgreSQL [1]. A wide variety of time series data was tested in their paper, from

which we took the financial data and electricity data to analyze in this section and section 4.5, respectively.

The financial data, NYSE Trade and Quote (TAQ), is obtained from Wharton Research Data Services (WRDS; https://wrds-www.wharton.upenn.edu/). TAQ contains intraday transactions data (trades and quotes) for all companies listed on the New York Stock Exchange (NYSE), American Stock Exchange (AMEX), and Nasdaq National Market System (NMS) and SmallCap issues. Stocks with average prices below 30$ across the available period and those with missing values were removed from the table for easy computation. Finally, stock prices of 839 companies from October 2004 to November 2019 were analyzed in this section. This means that each time series (stock price of one company) comprises of 3993 timepoints. For example, Figure 4-8 shows a company's stock price for the first 180 days.

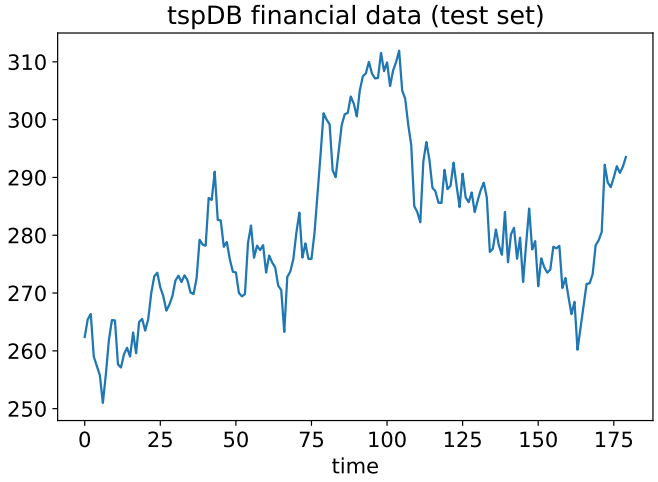

Figure 4-8: elec

We can observe that the rolling mean of the time series is not stationary, as it is expected for many other finance data sets. Therefore, a first order difference of the time series is calculated and plotted in Figure 4-9.

Since it is a sequence of continuous random variables, the data go through a pre-processing (section 2.2.4). Finally, after discretization, the two-step method can be applied to the modified sequence. Using this sequence of length $T = 179$, the error lower bound is tested for various choice of the number of bins (2^k).

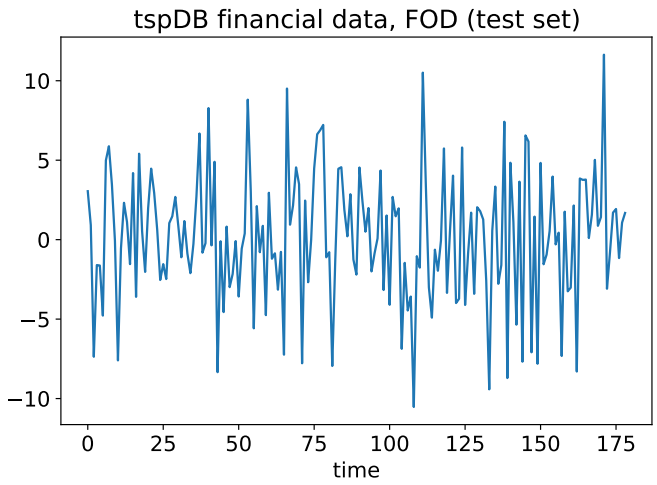

Figure 4-9: The first order difference of the financial data

In addition, three forecasting methods were adopted to make predictions: a Long-Short-Term-Memory (LSTM) neural network, DeepAR (industry standard deep learning library by Amazon), and a Real-Time Time Series Prediction System (TSPS, [1]). Following the standard goal in finance, the first 3813 time points for training and 1-step ahead forecast for 180 days were made and tested. Note that the prediction modeling was done with the original sequence, not discretized nor processed to obtain the first order difference. After the discretization, the prediction error was calculated in a discretized manner, i.e., by discretizing both the true value and the prediction, and comparing the two.

The result is shown in Figure 4-10—the red line denotes the error lower bound obtained by the two-step method and the three other colored lines are the prediction error rate for each model. The error lower bound increases as the number of bins increases. This is expected since it will be similar to predict the exact same number in a continuous scale (the probability of two random numbers in $\mathbb{R}$ to be the same is zero). For all choices of k (on the horizontal axis, the number of bins for discritization is 2^k), the actual prediction error for all three models are higher thatn the estimated lower bound.

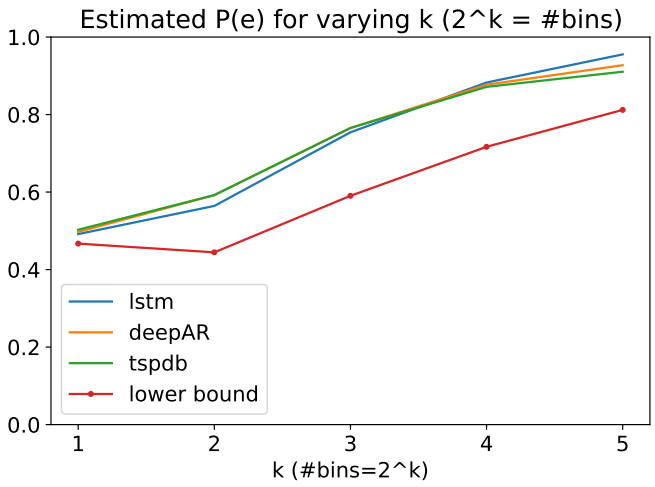

Figure 4-10: Prediction error lower bounds and actual prediction errors; financial data

4.5 Electricity Data

Similar to the section above (secion 4.4), we analyze electricity dataset in this section. The electricity data is from the UCI data repository (https://archive.ics.uci.edu/). The dataset contains the electricity usage of a household in kW per 15 minutes. The data is converted to kW per hour and one example is plotted in Figure 4-11. Similar to the earlier section, we take the first order difference and it is shown in Figure 4-12. Note that the seasonality observed in the original time series is still not removed when the first order difference was taken.

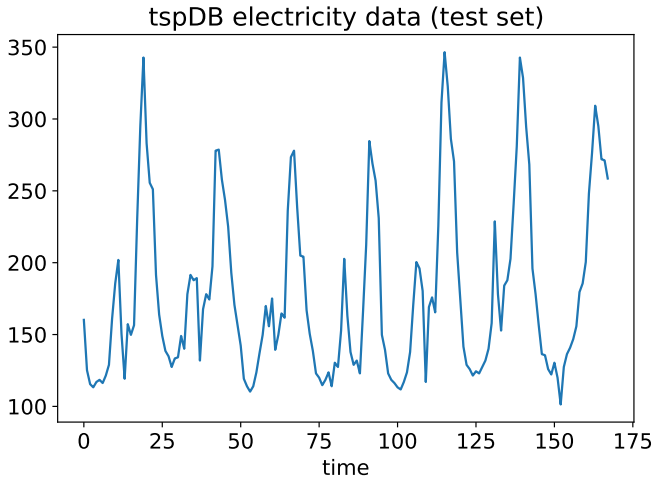

Figure 4-11: Electricity usage of a household

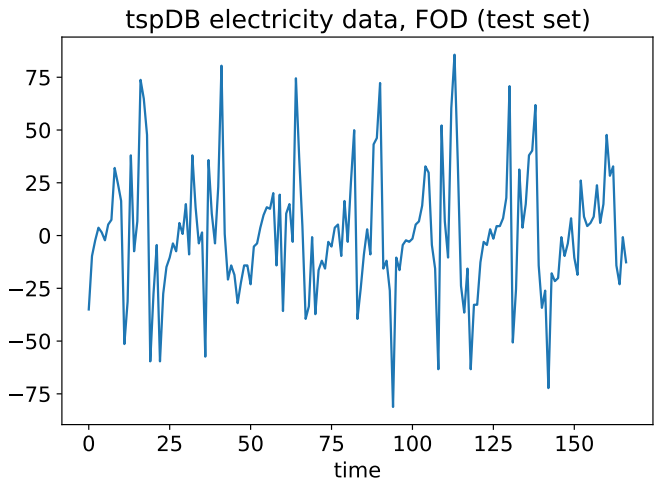

Figure 4-12: First order difference of the electricity data

Similar to what we did for the financial data, the three forecasting methods were adopted to make predictions: a Long-Short-Term-Memory (LSTM) neural network, DeepAR (industry standard deep learning library by Amazon), and a Real-Time Time Series Prediction System (TSPS, [1]). The first 25968 time-points are used for training; and day-ahead forecasts for the next seven days (i.e. 24-step ahead for 7 windows) are made and tested. Figure 4-13 shows a similar plot to Figure 4-10, however, the red line (lower bound) is not strictly below the prediction error lines.

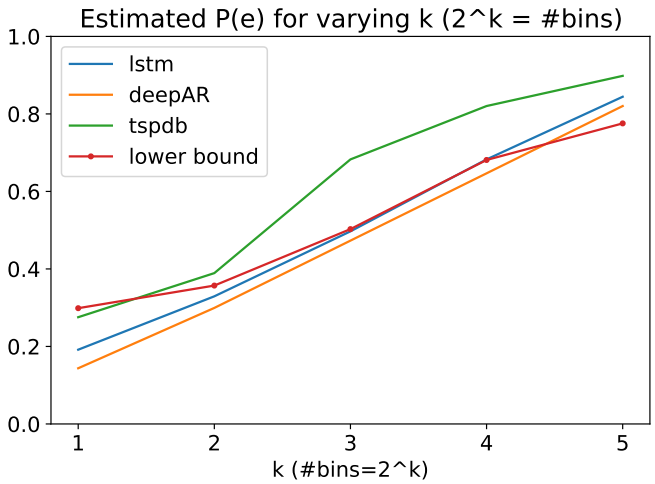

Figure 4-13: Prediction error lower bounds and actual prediction errors; electricity data

Chapter 5

Discussion

In this chapter, we discuss the results of this research. The first section goes over Chapter 4 and focuses on the results of the five demonstrations. The second section claims the contribution of this research in a higher level. The last section points out the limitations of the suggested two-step method. These will be highlighted again in the last part of this thesis to suggest future research directions.

5.1 Discussion on the Use of the Two-Step Method

In section 4.1, a discrete time series with discrete random variables was tested. The time series had four stages—1) quiet sleep, 2) indeterminate sleep, 3) active sleep, and 4) awake—and they do have a relative meaning to each other. As the numbered labels are ordinal, not nominal, it would have been more useful to obtain a lower bound of an error metric with a notion of distance, such as mean squared error.

The classification rate of the best performing model reported in the paper was 0.034, which is slightly lower than the estimated lower bound (0.0375). According to this estimation, their model could be seen as near-optimal. Several factors may contribute to the low error rate than the estimated lower bound. As mentioned earlier in section 3.1.3, the fundamental stochasticity of the random models and approximation error when applying the inverse function could be a reason. Furthermore, the modeling and testing procedure in the paper had only 322 measurements in the test set.

In section 4.2, a discretization method was borrowed from Amjad's paper. This definition of defining binary or ternary time series from continuous variables makes sense in the bitcoin or other financial datasets such as stock prices, as it is a common question for those who are interested in the issue to forecast if it will increase, decrease, or remain the same.

In section 4.3, we took the first order difference to make it reasonable to assume stationarity—it is hard to convince others that an ever-increasing time series is stationary. One interesting question that we can throw is what the increasing trend in the error lower bound estimation per season means. If we can say that the games are harder to predict these days, we may be able to claim that the games are more entertaining these days. Also, a smaller standard deviation could be construed as a sign of high performing athletes, trained with a more developed programs and assisted by technology, who rarely make mistakes.

In section 4.4 and section 4.5, financial data (stock price) and electricity data (home electricity usage) were analyzed and compared to the prediction models such as LSTM, DeepAR, and TSPS. As the data consist of continuous random variables, discretizing the distribution preceded the analysis. Although the two-step method works with the continuous data via discretization process, the efficacy of the obtained error lower bound is quite doubtable. In most cases, the probability of error larger than a certain distance (e.g.,$\mathbb{P}(|X - \hat{X}| \geq l)$) will be more appropriate than the error lower bound of a transformed sequence.

In addition, we have observed several cases where the empirical prediction error is lower than the estimated error lower bound. This is more evident in the electricity data, as well as in the sleep data. There are several factors that may have attributed to this result. For the electricity data, the prediction model was trained with the continuous data and then the prediction output was converted to a discretized sequence to calculate the prediction error in terms of the discretized distribution. As the prediction model had access the real sequence with full information, unlike the discretized sequence that has less juice in it, the prediction may have been easier than predicting the discretized bins out of the discretized observation.

Another reason could be the inherent stochasticity and approximation error of the method. We sample random sequences to learn the relationship between the compression ratio and the entropy rate. The sampling process inevidently ensue the error from its stochasticity that could be minimized by sampling more or increasing the length of the sequences. Also, the second step incorporates gradient descent method, which creates another source of the approximation error. These do not only apply to continuous random variables but also to already discrete random variables.

5.2 Contribution

In this thesis, an entropy rate estimation technique was proposed as a subroutine for error lower bound estimation process. By utilizing Lempel-Ziv algorithm, an accessible and light method to estimate the entropy rate of a time series was presented. The quality of prediction was validated with multinomial and Markov processes, showing a desirable performance for both independent (multinomial) and dependent (Markov) sequences. This can be not only used for the suggested way (to obtain the error lower bound) but also adopted in other situations where one needs to estimate the entropy rate of a discrete sequence that is stationary.

On top of the entropy rate estimation, Fano's inequality was employed to finalize the two-step framework for error lower bound estimation. The classification error lower bound was approximated via gradient descent, using the entropy rate estimation from the first step. The method is easily implementable using programming languages such as Python, and does not make the independency assumption, i.e., the data points may not be independent to each other.

Furthermore, this method can be useful to inform scientists and policymakers who want to utilize time series prediction algorithms to forecast the future. Once we apply the two-step method to the time series of interest, we can obtain the error lower bound that the original data generating source allows us to predict as it is based on the entropy rate of the time series. The proposed two-step method can be interpreted as "the best quality of estimation we could obtain" and help the policymakers decide

whether or not to adopt a new prediction algorithm or inform the scientists whether we have already reached the practical optimal or not.

5.3 Limitation

The suggested error lower bound is based on Fano's inequality and entropy rate estimation. Hence, there are some restrictions on the type of time series that are eligible as we have made some assumptions around the data source.

First, the stationarity of the time series is assumed when we used Theorem 3. In words, stationarity means that we assume that the underlying data generation process' characteristics do not change. However, this assumption is not easy to hold in most of real life situations. If there is a way to estimate the entropy rate at a specific time without assumping the stationarity, the application of this method could be wider than it is now. However, we have to calculate the entropy rate and estimated error lower bound for each timepoint in such cases.

Next, the proposed lower bound is only for the classification error. This automatically means that we can only apply the method to discrete random variables. Hence the best situation to apply this method is restricted to binary classification or nominal cases where the labels do not have an order or a relative proximity to each other. Nonetheless, the method can be applied to any discrete random variable if it satisfies other assumptions such as stationarity.

Lastly, the suggested method is only for a univariate time series. However, there are many multivariate time series out there and we can learn more about the stochastic process when auxiliary observation is available. It is natural to utilize other time series that may reveal more information about the sequence of interest, and usually this way can reach a lower prediction error. For those situations, however, the two-step method presented in this thesis is not applicable to assess the quality of prediction algorithm.

Chapter 6

Error Lower Bound as a Knowledge Assessment Tool

6.1 Knowledge Assessment

Knowledge is one of the main outcomes of science, and the impact of science on our society largely depends on the use and interpretation of scientific knowledge. Using scientific knowledge in the decision making process is a common way how science is involved in the lives of people, both for scientists and non-scientists. Knowledge assessment would have been less complicated if everything was not too dynamic, however, the world we are living in is full of uncertainties. Therefore, assessing the legitimacy of scientific knowledge under conditions of uncertainty and controversy is becoming more crucial these days.

Technocratic and adversarial knowledge assessment are two types of belief system about how we reveal the truth. The technocratic approach assumes that the neutral and honest scientist can act as an information broker, which we can rely on and consult with to find out the truth. On the other hand, the adversarial approach denies the existence of such "neutral" agent, and advocates that we should fight among often biased partisans. The key difference is the controversy over whether it is possible to have an insulated neutral agent.

6.2 Role of Technocratic Knowledge Assessment Tool

Scientists often describe scientific findings as a value-neutral discovery without any personal values laden under the argument, however, many science and technology researchers including Walker have confirmed in numerous areas that science can hardly be value-neutral [27]. Walker specifically discusses about the potential existence of a "neutral arbiter" for triggering precautions and contends that science cannot be such a neutral arbiter because scientific decision making involves non-scientific decisions as well. Especially in situations where the uncertainty is relatively high—and hence the degree of confidence is relatively low—, the scientist's values play a bigger role in defining risk and analyzing the future.

While admitting that it is almost impossible to have a perfectly neutral science arbiter, I want to argue that we should seek for the technocratic approach. To do so, we will review two cases, one good example where technocratic knowledge assessment worked out, and one bad example where technocratic approach did not seem to perform well.

One of the good examples where the technocratic approach was helpful is the research about the effects of second hand smoking. As second hand smoking became an issue, there were many research conducted with confounding findings—some concluded that it is not harmful, some said that there is not enough evidence to claim its toxicity, and others said it is very dangerous. Later on, one organization conducted a meta-level study on those research and revealed a strong correlation between the findings and the research funding source (funded by tobacco-related firms or non-tobacco-related firms/government). As one can imagine, most of the research funded by tobacco-related companies found the second-hand smoking not harmful, whereas a lot more non-tobacco-funded research concluded that it indeed has an adverse impact on health. I view this as a successful application of the technocratic approach, since we were able to clearly show the relationship between the funding source and research findings, which provided a grounded evidence on why we should believe the research that found the second-hand smoking harmful rather than others.

In this case, the most important condition was that a "neutral" organization could be established. It was free from the power relationship with the tobacco firms because the funding was directly coming from the government. I still acknowledge that the neutrality of this organization could become controversial, but at least it had a power to conduct the meta-level research without the tobacco firms hindering them. In addition, the research funding sources were identifiable so that the research team could conduct a meta-level investigation—otherwise it would have been just impossible to reveal the correlation. Lastly, there were a lot more research conducted by non-tobacco affiliated organizations. If there were 100 papers done by the tobacco-affiliated teams and only 10 by the non-tobacco affiliated teams, it could have been difficult to show the correlation in a clear and intuitive manner. However, the number of research papers published by non-tobacco affiliated organizations was significantly greater than the ones published by tobacco-related teams. Also, because most of those research papers had aligned opinions, the uncertainly around the issue was relatively low.

On the other hand, there are some cases where the technocratic approach fails to give us a fast lane to reach the truth. For example, the sciene community struggled for more than several decades to confirm the effect of low-level mercury exposure. There were two main studies conducted to verify the effect of low-level exposure to mercury on human body, especially on pregnant women and their babies. The two studies showed contradicting findings—one claiming that there is impact, the other saying there is no impact—and later there was a panel talk to discuss this issue. However, the panelists (scientists) could not confirm which side we should believe, and verified that the both studies are legitimate. This confounded the public even more and a huge adversarial controversy was backfired, including the ad on New Yorker with a tagline: *"Concerned about mercury? You shouldn't be. Unless you eat this."*— with a picture of a canned whale meat; the ad is in the Appendix, Figure B-1.

The biggest problem in this case is that the scientific research failed to persuade the public. First of all, the research was not relevant to most of the customers, including pregnant women. The panel talk only confounded the public by not giving a concrete "answer" to the question: is low-level exposure to mercury safe or not? Also, unlike the

former example, the uncertainty was very high as the research was hard to duplicate due to its inherent setting and methodology. While the scientists were failing to reach an agreement and persuade the public with the scientific knowldege, the New Yorker ad registered by the consumers' organization, stating that "you should eat half a whale to get that level of exposure to mercury and suffer from side effects," promoted an adversarial controversy over the topic.

The technocratic knowledge assessment is never a perfect approach to reveal the ultimate truth, yet the commitment to reliance on an expert system is desirable for us to extract the most out of science in under the uncertainties. We know that adversarial knowledge assessment cannot be stopped, and it is even "natural" to happen when we have no access to the truth. However, we should look for more technocratic approaches while admitting that the adversarial flow could always happen, and scientists are often biased as well. What we should do is not to abandon the technocratic approach, but to come up with the policies to prevent potential fallacies.

6.3 Error Lower Bound as a Technocratic Knowledge Assessment Tool

As a technocratic knowledge assessment tool, the suggested error lower bound and the two-step method to obtaining it can benefit the scientists who want to predict the future in a wide array of areas. The method can be applied to any discrete time series with discrete random variables, regardless of the domain. There are certain conditions and assumptions, such as stationarity and ergodicity, but they do not largely change the use of this method as a technocratic knowledge assessment tool. The calculated error lower bound incorporates the data source's inherent uncertainty so that the scientist does not need to weigh in their subjective opinion on the risk. In other words, the entropy is a fair way to capture the underlying distribution's uncertainty from the observation.

The suggested lower bound will signify the best that we can predict under this

observed level of uncertainty. I want to distinguish this from the feasible or obtainable lower bound. The error lower bound is calculate for a data source that generates time-series based on an unknown rule. The presented method estimates the entropy of the time-series by observing historic data. Using the entropy, it estimates the prediction error lower bound. Therefore, it does not guarantee that there exists a prediction algorithm that reaches this error lower bound. Yet, the user (who wants to use this method to obtain the error lower bound) can use this as one criterion to evaluate the current algorithm's performance by comparing the error rate of the algorithm of interest and the lower bound.

Chapter 7

Conclusion and Future Research Directions

7.1 Concluding Remark

This thesis was composed in the following order. In Chapter 1, the motivation for this research and the detailed problem setting was presented. In Chapter 2, background information to understand how the suggested method works was illustrated: the main focus was on the Lempel-Ziv compression and Fano's inequality. The two-step method was suggested and validated in Chapter 3, and demonstration of the method with real-world datasets were included in Chapter 4. In Chapter 5, we discussed how this method can be viewed as a technocratic knowledge assessment tool. The Chapter 6 summarizes the contribution and limitations of the research.

This research suggested an approach to estimate the prediction error lower bound of a stationary times series via entropy rate estimation. The entropy rate of discrete-valued sequences was conveniently calculated by building a compression ratio - entropy rate regression model, where the Lempel-Ziv algorithm was utilzed to compress data. Furthermore, this research suggests an way to apply the method to a real-valued data by discretizing the numbers into fixed-sized bins. This study demonstrated the use of this approach to several examples including both discrete and continuous time-series data. The utility of this error lower bound is assessed by adopting the Kalman filter

estimates, the optimal predictor for the Gaussian linear model.

7.2 Future Work

Based on the discussion about the limitations of this research, I would like to suggest three potential research directions: 1) removing the stationarity assumption, 2) expanding to obtaining the regression error, and 3) incorporating the multivariate time series.

First, one could devise a method without the stationarity assumption that can still estimate the prediction error lower bound. The suggest method requires the stationarity assumption because it learns the statistical behavior of the time series, i.e., entropy, by analyzing previously observed data. That being said, the method assumes that the entropy remains the same for the whole time (time of the observed history and times that we want to make a prediction). For real-world observation datasets, it is unnatural to expect it to be perfectly stationary. One way to go about it is to assume that the time series is stationary for a certain period of time. For example, even if we have a year-long observation, we could assume that the entropy will remain the same for 1-month interval. Otherwise, we can design a new approach to estimate the entropy at a specific moment, which will be the time that we want to make a prediction, not necessarily assuming the stationarity of the time series.

Next, one could adopt a differential entropy and expand this framework to incorporate continuous random variables—this will allow us to obtain the regression error lower bound. The proposed method is for discrete random variables, and the probability of error is a classification error, $\mathbb{P}(\hat{X} \neq X)$. This notion of prediction error works well for a binary classification or a nominal prediction problem. However, discrete random variables oftentimes have an ordinal meaning in its labels, such as intensity level or likert scale, and it is better to choose a regression error as an error metric for those time series. There could be many ways that can potentially open the doors for regression error lower bound estimation. One way is to adopt the notion of the differential entropy and develop a method to estimate the differential entropy

of a continuous-valued time series. In such cases, one should utilize an inequality with regression error, other than Fano's inequality which contains the probability of a classification error. For example, one could rewrite the Fano's inequality to have a notion of regression error by defining δ and $\mathbb{P}(|\hat{X} - X| \geq \delta)$ [11].

Lastly, one could work to extend this method to apply on a multivariate time series. Even if we are interested in just one time series, there are many auxiliary datasets that can potentially reveal some information about the sequence of interest. For example, if you are interested in predicting electricity consumption in a household, the time data (whether it be time of the year or of the day) may be able to hint your algorithm to perform better. In this case, if the algorithm utilizes multivariate time series, the proposed method is not applicable. As it is getting more common and common to have a multivariate time series than a univariate one, extending this work to apply on a multivariate time series is desirable.

Appendix A

Useful Links

A.1 Data used in demonstration

1. Sleep Data: http://www.mas.ucy.ac.cy/ fokianos/bookts.htm
2. NBA Data: https://stats.nba.com/
3. Financial Data: https://wrds-www.wharton.upenn.edu/
4. Electricity Data: https://archive.ics.uci.edu/

A.2 Github repository

1. Time series lower bound estimation code used in this thesis:
   https://github.com/saeyoung/tslb
2. Time series prediction DB, including the lower bound estimation tool:
   https://github.com/AbdullahO/tspdb

Appendix B

Figures

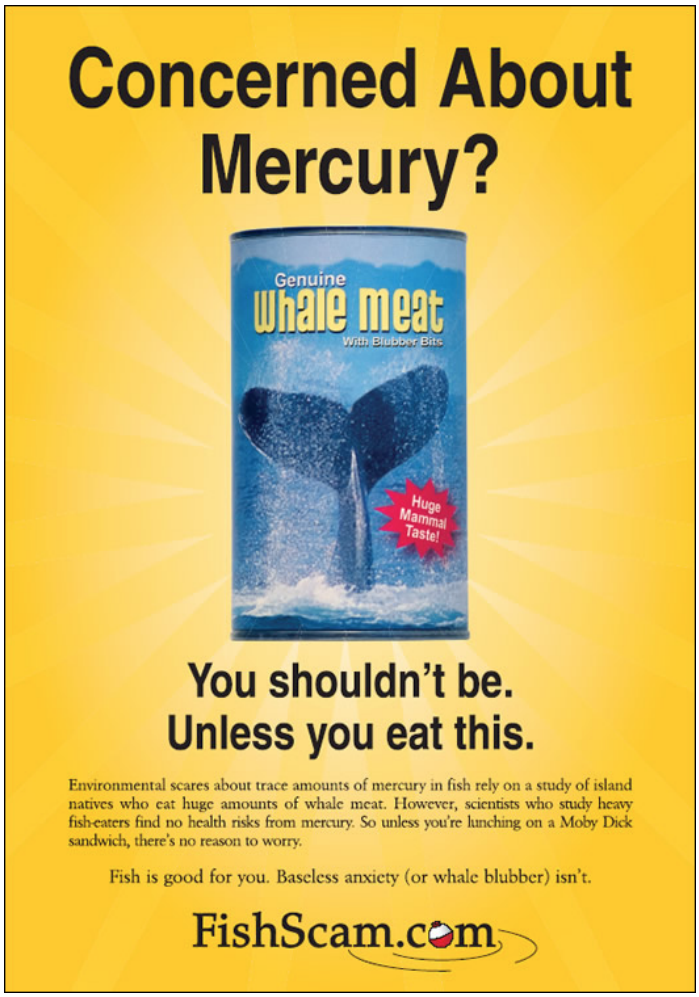

Figure B-1: A full-page ad in the New Yorker, April 2006.

Bibliography

[1] Anish Agarwal, Abdullah Alomar, Muhammad J Amjad, Robert Lindland, and Devavrat Shah. tspdb: Time series predict db. *arXiv preprint arXiv:1903.07097*, 2019.

[2] Anish Agarwal, Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. Model agnostic time series analysis via matrix estimation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(3):40, 2018.

[3] José M Amigó, Janusz Szczepański, Elek Wajnryb, and Maria V Sanchez-Vives. Estimating the entropy rate of spike trains via lempel-ziv complexity. *Neural Computation*, 16(4):717–736, 2004.

[4] Muhammad Amjad and Devavrat Shah. Trading bitcoin and online time series prediction. In *NIPS 2016 Time Series Workshop*, pages 1–15, 2017.

[5] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[6] Jenkins Box and Reinsel. *Time Series Analysis, Forecasting and Control*. Prentice Hall, Englewood Clifs, NJ, 3rd edition, 1994.

[7] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

[8] François Chollet. keras. `https://github.com/fchollet/keras`, 2015.

[9] Thomas M Cover. Behavior of sequential predictors of binary sequences. Technical report, DTIC Document, 1966.

[10] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[11] John C Duchi and Martin J Wainwright. Distance-based and continuum fano inequalities with applications to statistical estimation. *arXiv preprint arXiv:1311.2669*, 2013.

[12] Deniz Erdogmus and Jose C Principe. Lower and upper bounds for misclassification probability based on renyi's information. *Journal of VLSI signal processing systems for signal, image and video technology*, 37(2-3):305–317, 2004.

[13] Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *Information Theory, IEEE Transactions on*, 38(4):1258–1270, 1992.

[14] Konstantinos Fokianos and Benjamin Kedem. Regression theory for categorical time series. *Statistical Science*, 18, 08 2003.

[15] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[16] Yanjun Han, Jiantao Jiao, Chuan-Zheng Lee, Tsachy Weissman, Yihong Wu, and Tiancheng Yu. Entropy rate estimation for markov chains with large state space. In *Advances in Neural Information Processing Systems*, pages 9781–9792, 2018.

[17] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[18] Vladik Kreinovich, Hung T Nguyen, and Rujira Ouncharoen. How to estimate forecasting quality: a system-motivated derivation of symmetric mean absolute percentage error (smape) and other similar characteristics. 2014.

[19] Debra A Lelewer and Daniel S Hirschberg. Data compression. *ACM Computing Surveys (CSUR)*, 19(3):261–296, 1987.

[20] Reza Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7036–7042. IEEE, 2009.

[21] Jorma Rissanen. Universal coding, information, prediction, and estimation. *Information Theory, IEEE Transactions on*, 30(4):629–636, 1984.

[22] David S. Stoffer Robert H. Shumway. *Time Series Analysis and It's Applications*. Blue Printing, 3rd edition, 2015.

[23] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2019.

[24] Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

[25] Paul C Shields. The interactions between ergodic theory and information theory. In *IEEE Transactions on Information Theory*. Citeseer, 1998.

[26] Petr Tichavsky, Carlos H Muravchik, and Arye Nehorai. Posterior cramér-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on signal processing*, 46(5):1386–1396, 1998.

[27] Vern R Walker. The myth of science as a neutral arbiter for triggering precautions. *BC Int'l & Comp. L. Rev.*, 26:197, 2003.

[28] Kevin W Wilson, Bhiksha Raj, and Paris Smaragdis. Regularized non-negative matrix factorization with temporal dependencies for speech denoising. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[29] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.

[30] Yuchen Zhang, John Duchi, Michael I Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems*, pages 2328–2336, 2013.

[31] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.