Text Analytics of the Qur’ān using Bayesian Statistics and Large Language Models

by

Al-Ahmadgaid Bahauddin Asaad

Submitted to the *Institute of Islamic Studies*
in partial fulfillment of the requirements for the degree of
Master of Arts in Islamic Studies

University of the Philippines Diliman
Diliman, Quezon City

May 2025

Dedication

إلى والدي ألراحل، شكرا لك على كل شيء،

وعلى شعب فلسطين، وعلى الجيل القادم من هذه الأمة

To my late father, magsukul ma kamemon pa;

to the people of Palestine; and, to the next generation of this Ummah

أَفَلَا يَتَدَبَّرُونَ ٱلْقُرْآنَ وَلَوْ كَانَ مِنْ عِنْدِغَيْرِ ٱللَّهِ لَوَجَدُوا فِيهِ ٱخْتِلَافًا كَثِيرًا

Will they not think about this Qur'an? If it had been from anyone other than God, they would have found much inconsistency in it.

Qur'an 4:82

وَعِبَادُ ٱلرَّحْمَانِ ٱلَّذِينَ يَمْشُونَ عَلَى ٱلْأَرْضِ هَوْنًا وَإِذَا خَاطَبَهُمُ
ٱلْجَاهِلُونَ قَالُوا سَلَامًا

The servants of the Lord of Mercy are those who walk humbly on the earth, and who, when aggressive people address them, reply, with words of Peace.

Qur'an 25:63

Contents

List of Figures

Abstract

The interest of the paper is to provide a comprehensive text analytics of the Qur'an. This is by utilizing Statistical and Machine Learning methods to computationally analyze the said scripture. Specifically, the following procedures have been done: descriptive analyses of the structure of the Qur'an, morphological analyses of the Qur'an. Further, the data used for the Qur'anic Arabic corpus is the one in QuranTree.jl by Asaad (2022). The computational software used is Julia programming language.

Chapter 1

Introduction

The use of scientific computing to studying the Qur'ān is still in its early stage in the fields of Islamic Studies and Statistical and Machine Learning applications. This study will indeed benefit not only researchers from Islamic Studies but also Statisticians and Machine Learning researchers who are into text analytics. Having said that, it is therefore necessary to provide context to audiences of these disciplines to provide background on the state of Qur'ānic studies and the increasing adoption of scientific methodology to studying humanities.

1.1 Background

The Qur'ān or *al-qurʾān* اَلقُرآن meaning *the recitation*, the holy book of Islam, is revered by 1.9 billion (according to 2020 projection of Cooperman et al. (2011, p. 13)) Muslims across the globe as the literal words of God. Muslims believed that the Qur'ān was gradually revealed (Qur'ān 25:32) to Prophet Muhammad ﷺ through angel *ǧibrīl* جِبرِيل or Gabriel (Qur'ān 2:97). The Qur'ān contains 77,429 Arabic words in total, which covers only 56 percent of the Greek New Testament which has 138,020 words in total (Sinai, 2017, p. 11).

The Qur'ān is divided into *sūrahs* سُور which are the equivalent of chapters, each containing *ayāt* أَيَات (meaning *signs*), which are the equivalent of verses. The *sūrahs* سُور are not arranged in chronological order as in the Bible's books and chapters, but rather arranged in monotonically decreasing length of number of verses after the first *sūrah* سُورَة (*see* Figure 1.1). The *sūrah* سُورَة of the Qur'ān can be categorized into two types: the *makkiyya* مَكِّيَّة (Meccan) and *madaniyya* مَدَنِيَّة (Medinan). The categories refer to the geographical location of where the *sūrah* سُورَة was revealed. Figure 1.1 shows the grouping of the *sūrahs* سُور. Note that some of the *sūrahs* سُور have mixed geographical locations[1], that is, a few of the *ayāt* أَيَات in it were revealed in

[1] *see* list of the location in `https://tanzil.net/docs/revelation_order`

other geographical location apart from the geographical location of the rest of the *ayāt* أَيَاتْ. Therefore, the categorization in Figure 1.1 highlights the geographical location of the majority of the *ayāt* أَيَاتْ in the *sūrah* سُورَة.

The Qur'ān (meaning *the recitation*) was revealed *orally* by angel *ǧibrīl* جِبرِيل to Prophet Muhammad ﷺ and passed onto other believers through oral tradition (reciting the Qur'ān to students repeatedly so as to memorize it, instead of writing it down and let the believers read it and memorize it). Memorizing 77,429 Arabic words of the Qur'ān through oral transmission can be a difficult task, but what aids this memorization is the rhythm feature of the Qur'ān. According to one Orientalist, Sinai (2017), "rhyme, however, or rather a periodically recurrent word-final assonance, is a feature of the Qur'ān throughout, and it naturally partitions the

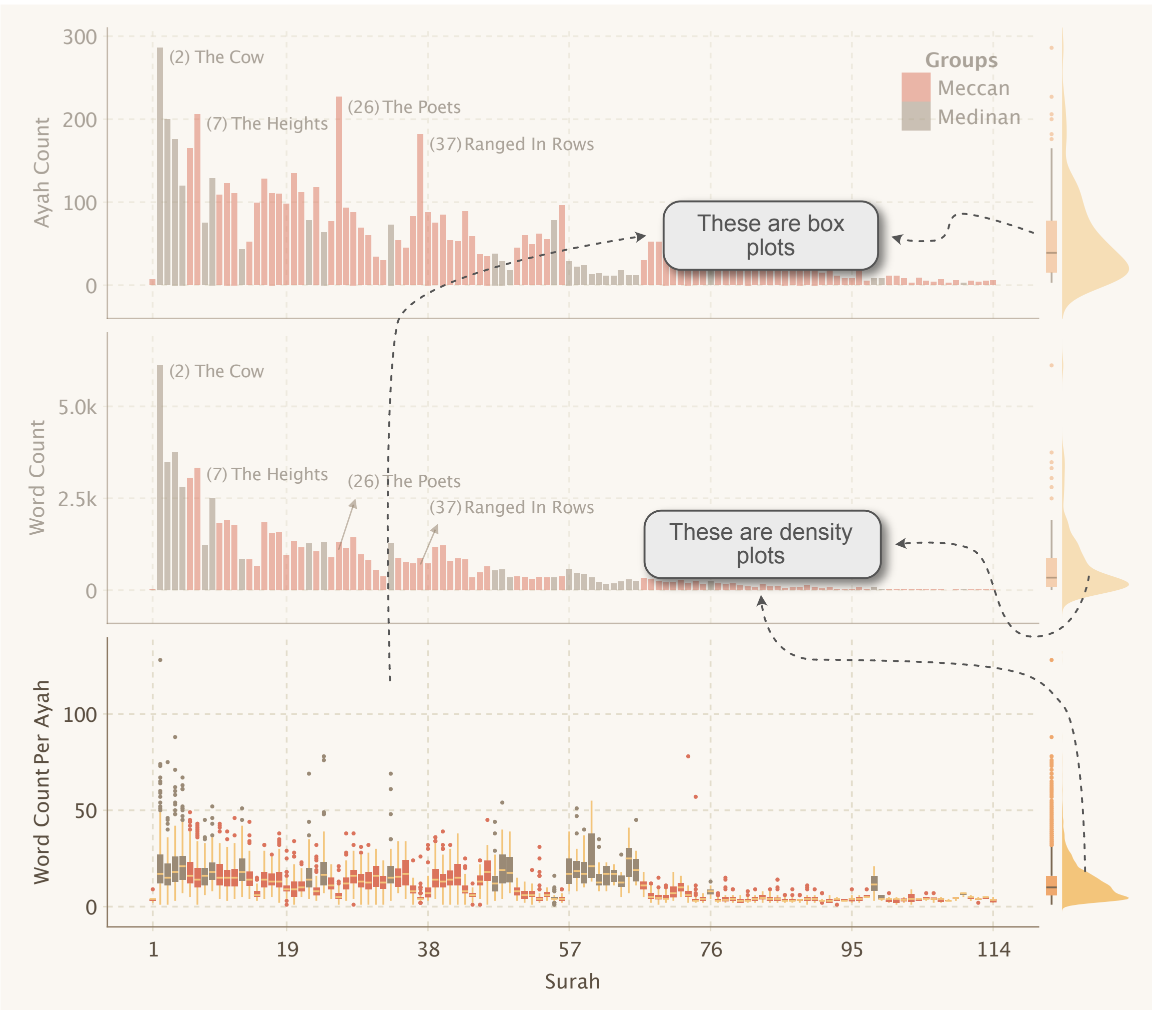


Figure 1.1: Statistics of the words and *ayāt* أَيَاتْ (verses) of the Qur'ān

sūrah سُورَة." Indeed, because of this feature, it makes it easy to memorize the entire Qur'ān, and the one who do so is called *hafiẓ* هَفِظ meaning *one who remembers* or *keeper*. Qur'ān memorization contest is a common event in Muslim countries, the Philippine embassy has hosted one in 2022 in Saudi Arabia (Manila Bulletin, 2022).

According to the Muslim tradition, the oral transmission of passing the Qur'ān from a *hafiẓ* هَفِظ to new believers was gradually put into writings as requested by the believers themselves. The idea was brought up after the battle of Yamama, where many of the Muslims who died were *qurrāʾ* قُرَّاء (*the one who properly recite the Qur'ān*), and so fearing that their numbers will reduce in other battle fields, Umar ibn al-Khattab عُمَرُ بنُ الخَطّاب (who became the second caliph) suggested to the first caliph, Abū Bakr 'Abd Allāh ibn 'Abī ibn 'Abī Quḥāfa, أَبُو بَكرٍ عَبدُ اللّاهِ إِبنِ أَبِي قُحَافَةَ or short for Abū Bakr أَبُو بَكر, to collect the Qur'ān into writing. Abū Bakr then authorized Zaid ibn Thabit زَيدِ بنِ ثَابِتٍ for the task. According to Zaid, he started collecting from the leafless stalks of the date-palm tree and from the pieces of leather and hides and from the stones, and from the chests of men (who had memorized the Qur'ān, i.e. the *hafiẓ* هَفِظ)[2]. Long story short, the effort was finally codified by the third caliph, Uthman ibn Affan عُثمَانُ بنُ عَفّان in the year 645 CE, which was then recopied and distributed to the different regional capitals of the early Islamic empire of that time. The rest of the copies outside this codification were then burned[3] down in order to have one standard Qur'ān. The Qur'ān nowadays is therefore assumed to be based on Uthmanic codex because of the story mentioned. That is, if indeed Uthman has ordered to burn other copies of the Qur'ān outside his codification, then what's left should only be based on his codex or archetype, and that should only be the inherited codex of the Muslims today.

The Qur'ān is believed by the Muslims to have been preserved since it was first recited by angel *ǧibrīl* جِبرِيل or Gabriel to the Prophet ﷺ. Many orientalists had been skeptic about this claim, for example, John Wansbrough theorized that the Qur'ān was collected over a 200-year period (*see* Wansbrough, 2004, p. 101) after the death

[2] *see* `https://sunnah.com/bukhari:7191`
[3] *see* `https://sunnah.com/bukhari:4987`

of the Prophet ﷺ, instead of within a few years after the death of the Prophet ﷺ. However, recent findings through radiocarbon dating brings forward strong evidence of potential preservation of the whole Qur'ān, which the Muslims believed to be so. For example, the Birmingham Qur'ān manuscript discovered in 2015 is dated to be between 568 and 645 CE with 94.5% accuracy, making it among the oldest Qur'ānic manuscript in the world (*see* Birmingham University, 2015). Its predicted range of years intersects with the lifetime (570 to 632 CE) of the Prophet ﷺ. What is interesting is that the Birmingham Qur'ān is consistent with the Qur'ān today, word-by-word and letter-by-letter[4], see Figure 1.2. This is indeed another evidence that the Qur'ān today was codified by Uthman since the discovery of the Birmingham Qur'ān manuscripts have confirm it. In addition to this, the Sana'a Palimphest is also among the oldest Qur'ān radiocarbon dated to be between 578 CE and 669 CE with 95% accuracy (Sadeghi & Bergmann, 2010), which according to Sinai (2017), "neither does the edited portion of the Sana'a palimpsest offer evidence for additional or missing verses or for a divergent verse order within the *sūrahs* سُور." Given these discoveries on the recent Quranic manuscripts, the claim of Wansbrough (2004, p. 101) is now untenable (*see* Sinai, 2014).

One likely reason as to why the scribes were able to preserve the Qur'ān in the two folios of the Birmingham Qur'ān manuscripts follows from the fact that the Qur'ān is firstly memorized before it was decided to be written. So that, the Topkapi manuscripts that contain 99% (only 23 verses missing)—dated around 701 to 750 CE (Karatay, 1962)[5], that is, around 69 to 118 years after the death of the Prophet ﷺ—of the Qur'ān today was likely partly written from memory. This is possible since the Qur'ān possesses a rhytmic feature (that aids with memorization) that naturally divides its verses or *ayāt* أَيَات, and that the Qur'ān memorization competition is still held to this day (as in the example of Manila Bulletin, 2022), apart from

[4]The orthography of the Arabic letters in the early days had no diacritics and were written in its basic consonantal skeleton. That is, Arabic orthography and grammar were in their nascent when the Qur'ān was revealed, and therefore has to adjust and catch up, to capture and preserve the proper recitation of the Qur'an.

[5]*see also* `https://corpuscoranicum.de/en/manuscripts/1977/page/1-410?sura=1&verse=1`

the fact that it needs to be recited (any chapter after the first chapter of the Qur'ān according to the choice of the worshipper) every prayer from memory. Bottomline, there are many avenues for Qur'ān recitation from memory, and these have helped in its preservation. Moreover, since there are no significant evidence of insertion or malicious intention on addition or revision in all of the extant Qur'ānic manuscripts so far, some Orientalists came up with other theories of insertions on the basis of the literary style of the Qur'ān, see for example Sinai (2017, p. 92), where verse 102 of *sūra l-ṣāffāt* سُورَة الصّافّات or The Chapter of *Ranged in Rows* is theorized as addition because it is longer compared to other verses in the said chapter, refer to Sinai (2017, p. 92) for his other reasonings. Nonetheless, the Qur'ān is indeed stable based on the extant manuscripts.

Furthermore, the vastness of the early Islamic empire meant that different Mus-

Figure 1.2: 20th Century Qur'ān (left) in its fully featured orthographies vs Birmingham Qur'ān dated between 568 and 645 CE (right) in its basic consonantal skeleton. Image from Wikipedia (2015).

lim regional capitals have covered populations with different Arabic dialects, and so to accommodate these differences, Muslims believed that there were seven variant readings of the Uthmanic codex. Variant readings are defined as different pronunciations of the same word, in this case seven Uthamnic Qur'ān for seven different pronunciations. The *ḥadīṯ* حَدِيث or *narration* comes from Ubayy ibn Ka'b أُبَيّ بِن كَعب who reported[6] that the Prophet ﷺ was near the tank of Banu Ghifar that *ǧibrīl* جِبرِيل or Gabriel came to him and said: "... Allah has commanded you to recite the Qur'ān to your people in *seven dialects*, and in whichever dialect they would recite, they would be right." Recent work of Sidky (2020) shows that the material evidence on the regional variants is in remarkable agreement with well-attested written variants documented in the traditional Muslim literature.

Muslim and non-Muslim scholars alike have been extremely interested in understanding the unique literary characteristics of the Qur'ān. As mentioned earlier, unlike other books like the Bible (arranged in chronological order), the Qur'ān does not follow any obvious organization. In addition to this, a *sūrah* سُورَة does not fit the exact definition of a chapter. Indeed, the name attached to a *sūrah* سُورَة is often decided as the unique entity mentioned in the said *sūrah* سُورَة, its main purpose is to help early Muslims distinguish which *sūrah* سُورَة they are talking about, this is contrary to the chapter name where the associated name is obviously the main topic of the chapter. Further, as described by Sinai (2017), "... the compositional unity of the long surahs located at the beginning of the corpus is anything but obvious: at least at first sight, they can appear a flit back and forth between different topics in a largely haphazard manner. This impression is not limited to Western readers: even pre-modern Muslim scholars have often approached their scripture as a quarry of unconnected verses and groups of verses that bear little intrinsic relation to what precedes and follows." It wasn't until Neuwirth (2007), that the compositional unity of the Qur'ān can be observed in tighter literary unities, as Neuwirth (2007) showed that the many of these texts display a tripartite structure and are often constructive

[6] source: `https://sunnah.com/muslim:821a`

around a narrative middle part (Sinai, 2017). Samples of the organizational style of the Qur'ān was shown in Sinai (2017, p. 88).

1.2 Rationale of the Study

Attempts at understanding the Qur'ān by Qur'ānic scholars were mostly done with the use of manual processes, that is, studying the scriptures by going through its content one-at-a-time manually. However, with the advent of computers, some researchers have started using it to aid in their study. The first known to have used computers for studying the Qur'ān was likely Rashad Khalifa in 1968[7], where he studied the significance of the mysterious initials at the beginning of some *sūrahs* سُور. Rashad uploaded the Qur'ān into his computer by transliterating the Arabic letters and other Qur'ānic orthographies into Roman letters and symbols that the computer can easily parse. This approach of using computers to find new insights is more common in the field of science, and it was new to the field of Qur'ānic studies.

Indeed, to proceed with the use of scientific computing, the Qur'ān will be treated as the data that needs to be analyzed using what is called Natural Language Processing (NLP), a branch of Machine Learning (ML) that aims to understand natural languages, such as Arabic. To instruct the computer to do Statistical analyses, ML, or NLP, one needs to use a *software application* or a formal language called *programming language*. There are several programming languages that the computer can understand. The popular one for researchers in the field of sciences are Python, R, and sometimes Julia (Bezanson et al., 2017). These programming languages will be used to construct instructions for computer. Therefore, if the data is the Qur'ān, then there should be a way to interface with it using any of these programming languages. Or alternatively, there should be a way to upload it into the chosen programming language and encode the Arabic letters into something that can be easily parsed by the computer, like what Rashad Khalifa did. Having said that, there are indeed some programming languages with libraries or packages for

[7] *see* `https://www.masjidtucson.org/quran/miracle/a_profound_miracle_sura68nun133.html`

interfacing with the Qur'ān, and this is true for Python, R, and Julia. For this study, Julia programming language since its library for interfacing the Qur'ān using the QuranTree.jl[8] has more features compared to those available in R and Python (*see* Asaad, 2021). QuranTree.jl is based on Tanzil[9] for the Qur'ānic Arabic texts, and Dukes and Habash (2010) for morphological annotation, which both libraries from R and Python do not have in terms of morphological annotation from Dukes and Habash (2010).

Now that the programming language for this paper is identified, next is to understand how Statistics and Machine Learning can help in studying the Qur'ān. Statistics is a branch of Science that aims to study features or characteristics of data generated from a random phenomenon. The findings of Statistical analyses can then be used to make conclusions or predictions of the general population of the data or general characteristics of the data. Machine Learning or ML, on the other hand, is a branch of Artificial Intelligence that heavily intersects with Statistics, albeit with distinct differences as well. Both Statistics and ML aims to characterize data by learning its features, but ML research aims on complex models that are often inspired by simpler models from Statistics. Therefore, one can think of Statistics as one of the fundamentals of ML.

One of the goals of Statistics and Machine Learning is to summarize all of the learned features into a hypothesized equation or hypothesized mathematical formula whose parameters or weights are optimized to capture the most characteristics of the data. It is called hypothesized equation since the researcher is the one who decide which family of equation best describe the characteristics of the data. The said equation is called a *model*. Therefore, a model is a generic entity that can be fitted or molded to capture the features of the data. Mathematically and in general, a model can be written as:

$$\hat{y} := h(x|\theta), \tag{1.1}$$

[8] *see* `https://alstat.github.io/QuranTree.jl/stable/`

[9] `https://tanzil.net/download/`

which is a mapping between the sets $\mathcal{X}$ and $\mathcal{Y}$ where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ through a hypothesized function h. The idea of modeling is to find the optimal value of θ in such a way it will minimize the error between the actual data (represented by y below) and the predicted one $\hat{y}$:

$$\varepsilon := y - \hat{y} = y - h(x|\theta). \tag{1.2}$$

To help understand the concept of modeling, and relate it to the fashion industry, which the author assumes most readers are familiar with, a model in a fashion industry is responsible for representing the characteristics of the target customers (in Eq. 1.1 this is y). Therefore, for a clothing company, they hire Asian models (in Eq. 1.1, this is $\hat{y}$ or $h(x|\theta)$) to target Asian customers (in Eq. 1.1 this is y). So that, when these models wore the clothes sold by the said company, the potential customer will more or less be able to relate to the model, and be able to imagine themselves wearing that same clothes as well, which help them incline to buying the said clothing. The model, therefore, does not necessarily have the looks of every target Asian customers, but at least in terms of height, skin tone, hair, and other common Asian features, the model will likely have it, or at least the difference is more or less minimal (in Eq. 1.1, the difference is represented by $\varepsilon := y - \hat{y}$). The question now is, what are the benefits that this model can bring to the clothing company? Well, the clothing company will be able to create products that are tailored to their Asian customers using the said model, since the company will have the right baseline measurements needed. Relating this analogy to the technical concept discussed earlier, you can think of the target customers as the real or actual data (in Eq. 1.1 this is y), and the model as the same technical term use in Machine Learning and Statistics (in Eq. 1.1 this is $h(x|\theta)$), but this time this technical model is expected to capture the characteristics of the real data analogous to fashion model that is expected to capture the characteristics of the target customer. This Statistical or Machine Learning model brings the following benefit: researchers will be able to study the real data by simply using the model to answer questions that are not available in the sampled real data.

In this paper, the aim is to make use of the Statistical or Machine Learning modeling to understand the characteristics of the Qur'ān in terms of the morphologies of its canonical text in its modern orthography. More specifically, the following are the general objectives:

1.3 Objectives

The following are the general and specific objectives of this paper:

1. What are the structural characteristics of the Qur'an that can be extracted from its rich morphologies using statistical and large language models?
   (a) What are the statistics of the Qur'ān's morphological features in terms of its parts of speech and selected entities like God's name and the prophets names mentioned?
   (b) How do the rhythmic signatures of the Qur'ān of the verses looks like and what are statistical insights that can be extracted?
   (c) What are the supporting statistical insights on the current divisions of the Qur'ān's verses?
2. What other insights that can be extracted from the semantics of the Qur'an's texts using statistical and large language models?
   (a) How does the theory of *concentrism* be formulated statistically, and what are the insights from the statistical and large language models on this?
   (b) How do the surahs are organized in terms of the topics? What are the themes that can be extracted for each of the surahs?
   (c) How do these extracted themes compare to the summaries of Abdel Haleem's English translation of the Qur'an?
3. How does this combination of statistical, machine learning, and artificial intelligence with the Muslim's traditional literatures help in understanding the Qur'ān, especially with the advent of Generative AI?

1.4 Significance of the Study

While the Qur'ān has been extensively studied by Muslims and non-Muslims scholars alike, especially in the topic of Meccan and Medinan surahs, there is still a lot to uncover from the perspective of Computational Statistics. Hence, the significance of this study is that it brings forward new ways of extracting insights from the Qur'ān by leveraging Computations, Statistics, Machine Learning, and AI, that is still in its early stage in the field of Qur'ānic Studies. Therefore, this new perspective or process of studying the scripture not only aids the scholars of the Islamic Studies, but may also contribute indirectly to community development and policy makers who use Qur'ān as part of their decision making.

1.5 Scope of the Study

The paper will cover all chapters of the Qur'ān both for the Morphological and Topic Modeling analyses, but it will only present the results of *sūrahs* سُور with at least 1000 words. The rest of the result will be part of the web application that can be used to query the Qur'ān. It will also not delve into the *tafsir* تَفسِير of each of the verses, but only when necessary for further context.

1.6 Thesis Organization

The paper is organized as follows: Chapter 2 will discuss the related literatures, Chapter 3 will discuss the methodology, Chapter 4 will present the results and discussions, and finally Chapter 5 will contain the conclusion and recommendation. The references and appendices are placed after the Chapter 5.

Chapter 2

Literature Review

As mentioned in the previous chapter, the earliest work in Qur'ānic studies using computer was likely the work of Rashad Khalifa in 1968[1], which led to one of his book entitled 'The Computer Speaks: God's Message to the World' (*see* Khalifa (1981)). While the work of Rashad started at studying the mystery letters in the beginning of some *sūrahs* سُور (for example Qur'ān 2:1, 3:1, 7:1, etc.), it quickly went on to cover what he calls other *mathematical miracles*, all of which are covered in Khalifa (1981). His findings led him to generalize the claim that God revealed His words through this mathematical patterns throughout the Qur'ān, and that those verses that were off and did not conform to this discovered mathematical patterns led him to extensive investigation of the said verses, and concluded that those could be or surely be an insertion that should not have been in the Qur'ān in the first place. There are two verses that were off according to Rashad, and he called these verses as *false verses*[2], these are the last two *ayāt* أَيَات of *sūra l-tāwba* سُورَة التَّاوبَة or The Chapter of *Repentance*. These two verses were removed in Rashad's Qur'ān translation[3]. Rashad believed so much on his findings that he self-proclaimed himself to be a messenger[4] with this new findings and that the Qur'ān nowadays should conform to his found mathematical patterns. This self-proclamation led to his assassination.

Fast forward to 20th century, among the pioneers to creating a stemmer systen for the Qur'ān is the work of Thabet (2004). A stemmer system is a system for trimming inflected words into its basic form, which grammatically mean its the root form. For example, in English language the root word for *computational*, *computer*, *computation*, and *computerize* is *compute*. Therefore, from the root of the word forms different stems representing the different words. Hence, the idea of stemming is to

[1] https://www.masjidtucson.org/quran/miracle/a_profound_miracle_sura68nun133.html
[2] https://submission.org/App24.html
[3] https://www.masjidtucson.org/quran/frames/
[4] https://www.masjidtucson.org/submission/faq/rashad_khalifa_summary.html

trim these words into its basic form, so that it would be easy to do word clustering or grouping through word similarity. According to Thabet (2004), the rich morphology of the Qur'ānic language or the Classical Arabic makes it even more difficult to do word stemming. Moving on, Thabet (2005) builds on this stemming system, and used it for tokenization of the Qur'ānic words. Tokenization is the process of listing all of the words in a sentence, and Thabet (2005) makes use of Thabet (2004) to further cleanse the noise from these tokens brought by the morphological variants of the Classical Arabic words. After cleansing the data, Thabet (2005) makes use of a statistical methodology for clustering or grouping the chapters of the Qur'ān, in particular the statistical methodology used is the Agglomerative Hierarchical Clustering based on the Euclidean distance of the adjusted word frequency of the *sūrah* سُورَة.

The work of Thabet (2004) and Thabet (2005) makes use of the Qur'ān corpus transliterated to Roman letters and symbols. Indeed, with interest growing on studying the Qur'ān from the lense of Data Analysis and Natural Language Processing, more work have been put in place into creating digital corpi of the Qur'ān that captures the different aspects of its linguistic styles. Hence, a series of work by Sharaf and Atwell led to the following publications: Sharaf and Atwell (2009) studied knowledge representation of the Qur'ānic verb valences using FrameNet frames, the output of which is a lexical database of the corpus of Qur'ān verbs. Further, the work of Sharaf and Atwell (2012a) came up with corpus for the annotations of the Qur'ānic pronouns, the authors named it as QurAna. Building on this work, Sharaf and Atwell (2012b) came up with a corpus for studying Qur'ānic relatedness based on the commentary of Ibn Kathir ابن كَثِير, the authors named this corpus as QurSim.

Moving on, an unpublished work by Nassourou (2011) used a Support Vector Machine (SVM) to predict the classification of the place of revelation of the Qur'ānic *sūrahs* سُور. The idea was to use *sūrahs* سُور with well attested place of revalation as the training set, and then train an SVM to predict the remainings of the chapters.

Furthermore, the work of Shahzadi, ur Rahman, and Sawar (2012) developed a simple classifier based on the frequency distribution of words in *ayāt* أَيَات and the corresponding words in *sūrahs* سُور.

Moving on, the work of

Chapter 3

Background on Probability and Statistics

This chapter will discuss some statistical concepts that will be used to understand and build up the ideas behind the methodology of this paper, which is presented in the next chapter. With that said, the discussions are organized as follows: Section 3.1 will present the concept of Descriptive Statistics; Section 3.2 will discuss the Probability Theory; and, Section 3.3 will discuss the Statistical graphs or plots.

Further, the topics discussed here may be self-explanatory for Statisticians, ML researchers or those with Mathematical background. However, for the benefit of readers coming from humanities background, the paper will present the methodology as follows: mathematical formulas are formalized for purpose of brevity through Definition, Proposition, and Corollary, but immediate to these are explanations or examples aimed to be simple enough for non-statistician readers. As a guide, statisticians or ML researchers can simply read the Definition, Proposition, and/or Corollary. Whereas for humanities readers, the reading shouldn't not stress too much on the said mathematical formalities, and instead proceed to the discussions or examples immediate to it to aid with the understanding.

3.1 Descriptive Statistics

Among the basic statistical methodologies for summarizing information or data is what is known as Descriptive Statistics. From the name itself, these statistics are meant to convey simple descriptions of the data. For example, *mean* and *variance* are common statistics used for describing the data. The formulas for these statistics are given in the following definitions:

Definition 3.1.1 (Mean)**.** Let $x_i, i \in \{1, \cdots, n\}$ where $n \in \mathbb{N}$, then the *mean* of x_is is defined as follows:

$$\bar{x} = \sum_{i=1}^{n} x_i, \qquad \text{where } x_i \in \mathbb{R}. \tag{3.1}$$

Definition 3.1.2 (Variance)**.** Let $x_i, i \in \{1, \cdots, n\}$ where $n \in \mathbb{N}$ and let $\bar{x}$ be the mean defined in Defn. 3.1.1 then the *variance* of x_is is defined as follows:

$$\mathbb{V}\mathrm{ar}(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \qquad \text{where } x_i \in \mathbb{R}. \tag{3.2}$$

Definition 3.1.3 (Standard Deviation)**.** Let σ^2 be the variance, then the standard deviation, denoted by σ, is defined as $\sigma := \sqrt{\sigma^2}$, that is, square root of the variance.

The mean is simply the average of the data points, while the variance is a single number that measures or summarizes the distances of the data points from the mean. The variance therefore measures how spread or varied the data points are. In practice, however, standard deviation is more popular for measure of variability since it doesn't get big easily due to the square root operator. Standard deviation is also the preferred metric for this paper.

3.2 Probability Theory

Statistics is built around the concept of Probability Theory. Hence, it is important to understand how this mathematical theory behind the statistical concepts work. Probability theory is a discipline in itself. It is a branch of mathematics that aims at studying realizations or observations from random phenomena. It does this by using algorithms and models (*see* discussion in Chapter 1 for understanding the model). These probability models are mathematical formula design to characterize or describe the patterns of data points. The simplest form of these models is the univariate probability density functions. However, before discussing these models, the idea behind it should be build up from ground up so that humanities readers can appreciate it. To do this, the discussion will proceed with the concept of probability.

Definition 3.2.1 (Probability)**.** A probability is a mathematical concept that measures the likelihood or chance of some event to happen.

Indeed, this idea of probability is well known or is easily understood by many.

So that, the probability that someone will die is 1, meaning 100%, since that's how natures and biology work. Further, the probability of selecting one *ayāt* أَيَات from *sūra l-fātiḥati* سُورَة الفَاتِحَت through a random draw is one out of seven, assuming each *ayāt* أَيَات has equal chances of being drawn.

Now, to gradually formalize the concept into mathematics, the succeeding definitions will be discussed to build up the definition of a probability distribution.

Definition 3.2.2 (Sample Space)**.** Let $\omega_1, \cdots, \omega_n, \forall n \in \mathbb{N}$ be the list of all possible outcomes of a random event or a phenomena, then the sample space, denoted by Ω, is defined as the collection of all these possible outcomes including the empty set denoted by $\emptyset$. □

Example 3.2.1. Consider the random phenomenon or event where someone randomly picks a verse from the Qur'ān, what is the probability that it will be a Meccan مَكِّيَّة or a Medinan مَدَنِيَّة verse?

The possible output for such random event is either Meccan or Medinan. Suppose, we let Meccan to be denoted by ω_1 and Medinan to be denoted ω_2, then the sample space, which is the collection of all possible output is denoted as follows:

$$\Omega := \{\text{مَكِّيَّة}, \text{مَدَنِيَّة}\} = \{\text{Meccan}, \text{Medinan}\} = \{\omega_1, \omega_2\} \tag{3.3}$$

It should be understood that $\emptyset$ is also included in Ω, but is not written for brevity.

□

Definition 3.2.3 (Event)**.** Let $\Omega = \{\omega_1, \cdots, \omega_n\}, \forall n \in \mathbb{N}$, be the sample space, then an event, denoted here as $\mathscr{A}$, is defined as the subset of the sample space, i.e., $\mathscr{A} \subseteq \Omega$. □

Example 3.2.2. From Ex. 3.2.1, consider drawing two samples from the Qur'an, what is the sample space and give an example of a possible event?

Solution: The sample space is given below:

$$\begin{aligned}
\Omega =&\{(\text{مَكِّيَّة}, \text{مَكِّيَّة}), (\text{مَكِّيَّة}, \text{مَدَنِيَّة}), (\text{مَدَنِيَّة}, \text{مَكِّيَّة}), (\text{مَدَنِيَّة}, \text{مَدَنِيَّة})\} \\
=&\{(\text{Meccan}, \text{Meccan}), (\text{Meccan}, \text{Medinan}), && (3.4)\\
&(\text{Medinan}, \text{Meccan}), (\text{Medinan}, \text{Medinan})\} && (3.5)\\
=&\{(\omega_1, \omega_1), (\omega_1, \omega_2), (\omega_2, \omega_1), (\omega_2, \omega_2)\} && (3.6)
\end{aligned}$$

So that, if $\mathscr{A}$ is the event of drawing two *ayāt* أَيَات from the Qur'ān, then a possible event is given by

$$\mathscr{A} := \{\text{مَدَنِيَّة}, \text{مَكِّيَّة}\} = \{\text{Medinan}, \text{Meccan}\} = \{\omega_2, \omega_1\} \tag{3.7}$$

Therefore, $\mathscr{A} \subseteq \Omega$, read as $\mathscr{A}$ is a subset of Ω. □

Now, going back to the discussion on the concept of probability above and reflect on the example given, that the probability that someone will die is 1; and that the probability of selecting one *ayāt* أَيَات from *sūra l-fātiḥati* سُورَةُ الفَاتِحَةِ through a random draw is one out of seven. It therefore begs the following questions: how does one solve this? Like how does it translate into a formal mathematical computation?

Indeed, to appreciate the motivation of the succeeding definitions, it is important to devise a logical approach to computing probability mathematically, and this should explain why the following definitions are defined the way they are.

Probability as already defined in Defn. 3.2.1 is a measure, which will be formalized in Defn. 3.2.5. Indeed, this is analogues to measuring an object's size using a tape measure. So, when someone attempts to measure an object's size, there are conditions for the space of the object to be measurable. The first expectation is that, the space or area can be measured in several ways. Either measuring it as a whole, or measuring it piece by piece from its partitions. Now, measuring it as a whole using tape measure should be straightforward. However, measuring it by pieces can have several cases, and all of these should end up to the same measuring. These cases accounts for the fact that when dealing with pieces of the surface one can start with different sizes of the pieces to be measured. So that, all the collections of all these possible configurations of pieces are mathematically called σ-algebra, and this collection should include the following:

1. The 0 size piece, that is, the object's measurement should start at 0;
2. The remainings of the pieces given the pieces already measured;
3. The union of all the pieces.

The above explanation for the σ-algebra is condensed in to the following mathematical definition:

Definition 3.2.4 (σ-algebra)**.** Let $\Omega := \{\omega_1, \cdots, \omega_n\}, \forall n \in \mathbb{N}$, then the collections of all disjoint partitions, which are the events, are defined as the σ-algebra or σ-field, denoted here as $\mathfrak{F}$, and should satisfy the following conditions:

1. The empty set $\emptyset \in \mathfrak{F}$
2. If $\mathscr{A} \in \mathfrak{F}$, then the complement $\Omega \backslash \mathscr{A}$ is also an element of $\mathfrak{F}$
3. If $\mathscr{A}_1, \mathscr{A}_2, \cdots$ is a countable sequence of sets in $\mathscr{A}$, then the $\bigcup_{i=1}^{\infty} \mathscr{A}_i$ is also an element of $\mathfrak{F}$

□

Example 3.2.3. Given the sample space $\Omega := \{\text{Meccan}, \text{Medinan}\}$, the σ-algebra is

$$\begin{aligned}\mathfrak{F} = \{&\text{Meccan}, \text{Medinan}, (\text{Meccan}, \text{Medinan}),\\ &(\text{Meccan}, \text{Meccan}), (\text{Medinan}, \text{Meccan}),\\ &(\text{Medinan}, \text{Medinan}), \emptyset\}\end{aligned} \tag{3.8}$$

□

Definition 3.2.5 (Probability Measure)**.** Let Ω be the *sample space*, $\mathfrak{F}$ be the σ-algebra, p be the probability measure, then the probability of a set $\mathscr{A}, \mathscr{A} \in \mathfrak{F}$, on the probability space $(\Omega, \mathfrak{F}, p)$, denoted by $p(\mathscr{A})$, satisfies the following properties:

1. Non-negativity: For any set $\mathscr{A}, p(\mathscr{A}) \geq 0$
2. Normalization: $p(\Omega) = 1$
3. Countable additivity: For any sequence of disjoint sets $\mathscr{A}_1, \mathscr{A}_2, \cdots$ in $\mathfrak{F}$, such that the union $\bigcup_{\forall i \in \mathbb{N}} \mathscr{A}_i$ is also in $\mathfrak{F}$, we have: $p\left(\bigcup_{\forall i \in \mathbb{N}} \mathscr{A}_i\right) = \sum_{\forall \in \mathbb{N}} p(\mathscr{A}_i)$

□

Basically, the idea of Defn. 3.2.5 is to define the concept of ”measure” in general sense, although above is a special measure called probability measure. As before, one can think of this probability measure like a tape measure in simple sense. This tape measure has some properties that should be expected for a measurement tool. The first property or condition is that the probability measure has to be positive. Indeed, if we use any tape measure for measuring length, never will someone get a negative measure like -2cm length. It always has to be positive. Further, the second condition to expect is that for this type of tape measure called probability measure, the total measure of all objects available in the given space should be equal to 1. That is, the total is normalized to 1. Think of this like 100% coverage if we measure all of the objects. Lastly, for any object, this tape measure should be able to measure the object through partitions, such that the measure of the union of these partitions is equal to the sum of the measure of each partition. All of these conditions are logical criteria for a general idea of ”measure,” although the normalization part above is unique for probability measure. Note that, the concept of proability measure here should not be confined to measuring length as in the analogy, it should generalize to measuring volume and complex objects in general.

Definition 3.2.6 (Random Variable)**.** Let Ω be the sample space, and $\mathbb{R}$ be the set of all real numbers, then a random variable X is a function defined as $X : \Omega \rightarrow \mathbb{R}$. □

Example 3.2.4. Consider the example of drawing a random verse or *ayāt* أَيَات again, what is the probability that it will be an *ayāt* أَيَات from *sūrah l-baqara*'s *ayāt* آيَةُ سُورَة البَقَرَة or the Chapter of Cow?

The answer to this is 4.59% probability, this is because there are 286 *ayāt* أَيَات and there are 6236 verses in the Qur'ān, so that the probability is $\frac{286}{6236} = 0.04586$. The assumption here is that all of the *ayāt* أَيَات in the Qur'ān have equal chances of being picked up or drawn. □

Example 3.2.5. To apply the concept so far, consider again Ex. 3.2.4, what is the probability of getting 5 *sūrah l-baqara*'s *ayāt* آيَةُ سُورَة البَقَرَة if we randomly pick 20 *ayāt* أَيَات in total from the Qur'ān?

Solution. The following are given:

- $n = 20$ independent trials of drawing $x = 5$ أَيَات from the Qur'ān
- Each trial has two possible outcomes: *naʿam* نَعَم meaning Yes or *lā* لَا meaning No. That is, if the *ayāt* أَيَات is from the *sūrah l-baqara* سُورَة البَقَرَة then its نَعَم, otherwise لَا.

Now, the sample space consists of all possible sequences of 20 نَعَم and لَا. That is,

$$\Omega = \{(\text{لَا}, \text{نَعَم}, \text{لَا}, \cdots, \text{نَعَم}), \tag{3.9}$$

$$(\text{لَا}, \text{نَعَم}, \text{لَا}, \cdots, \text{لَا}), \tag{3.10}$$

$$\vdots \tag{3.11}$$

$$(\text{لَا}, \text{لَا}, \text{لَا}, \cdots, \text{لَا})\}. \tag{3.12}$$

In total, there are $20^2 = 400$ possible samples in the sample space Ω. Further, from Ex. 3.2.4, the probability of getting a *sūrah l-baqara*'s *ayāt* آيَةُ سُورَة البَقَرَة is 0.0459 or 4.59%. Therefore, if X is the random variable of an event of drawing an *ayāt* from the Qur'ān, if we assign لَا and نَعَم as either 0 or 1, respectively, then this would imply that mathematically $p(X = 1) = 0.0459$. In addition, the probability of getting an *ayāt* أَيَات from other *sūrah* سُورَة would be

$$p(X = 0) = 1 - p(X = 1) = 1 - 0.0459 = 0.9541. \tag{3.13}$$

Further, let Z be the random variable for the event of getting n نَعَم from 20 trials, then the problem is now equivalent to finding the number of ways to choose n possible positions out of 20 in the collection or set. This can be solved using the *combination* formula as shown below:

$$\binom{n}{x} = \frac{n!}{r!(n-r)!} \Rightarrow \binom{20}{5} = \frac{20!}{5!(20-5)!} = 15,504. \tag{3.14}$$

That is, there are 15,504 possible cases of 5 positions' configuration for نَعَم in a 20 trial. Moreover, in each of these samples 5 has a probability of $p(X = 1) = 0.0459$, while the remaining 15 has a probability of $p(X = 0) = 0.9541$. So that,

$$\begin{aligned} p(Z = 5) &= 184,756 \times p(X = 1)^5 \times p(X = 0)^{20-5} \\ &= 15,504 \times 0.0459^5 \times 0.9541^{20-5} \\ &= 0.0016. \end{aligned} \tag{3.15}$$

Hence, there is a 0.16% probability of getting 5 *sūrah l-baqara*'s *ayāt* آيَةُ سُورَةَ البَقَرَة when randomly drawing 20 أَيَات from the Qur'ān. □

Note that Ex. 3.2.5 can be solved using a known formula in probability called *Binomial* mass function, which is a model that can be used to describe the event of getting 5 *sūrah l-baqara*'s *ayāt* آيَةُ سُورَةَ البَقَرَة on a 20 random samples of Qur'ān's أَيَات. The Binomial mass function is specifically a probability mass/density function model. The following section will discuss the Statistical Graphs, which will cover the concept of frequency distribution, the one modeled or characterized by the probability mass/density function.

3.3 Statistical Graphs

Graphs or plots are data visualization tools that are useful for exploratory data analysis apart from the Descriptive Statistics discussed above. It supplements the Descriptive Statistics findings through shapes visualized in the graphs. Among the popular statistical graphs is the bar graph. An example of this is given in Figure 1.1. Other graphs used are the box plots and the density plots which is also in Figure 1.1

3.3.1 Box, Density, and Histogram Plots

While most statistical plots are easy to understand like bar graphs and scatter plots, others like Box, Density, and Histogram plots may not be easy to comprehend for someone with no Statistical background. This section will discuss how it is interpreted. Let's use Figure 1.1, Figure 3.1 for easy reference in this section.

As shown in Figure 3.1, both the Box and Density plots are tied to each other. This is indeed the case because both are describing the same information but pre-

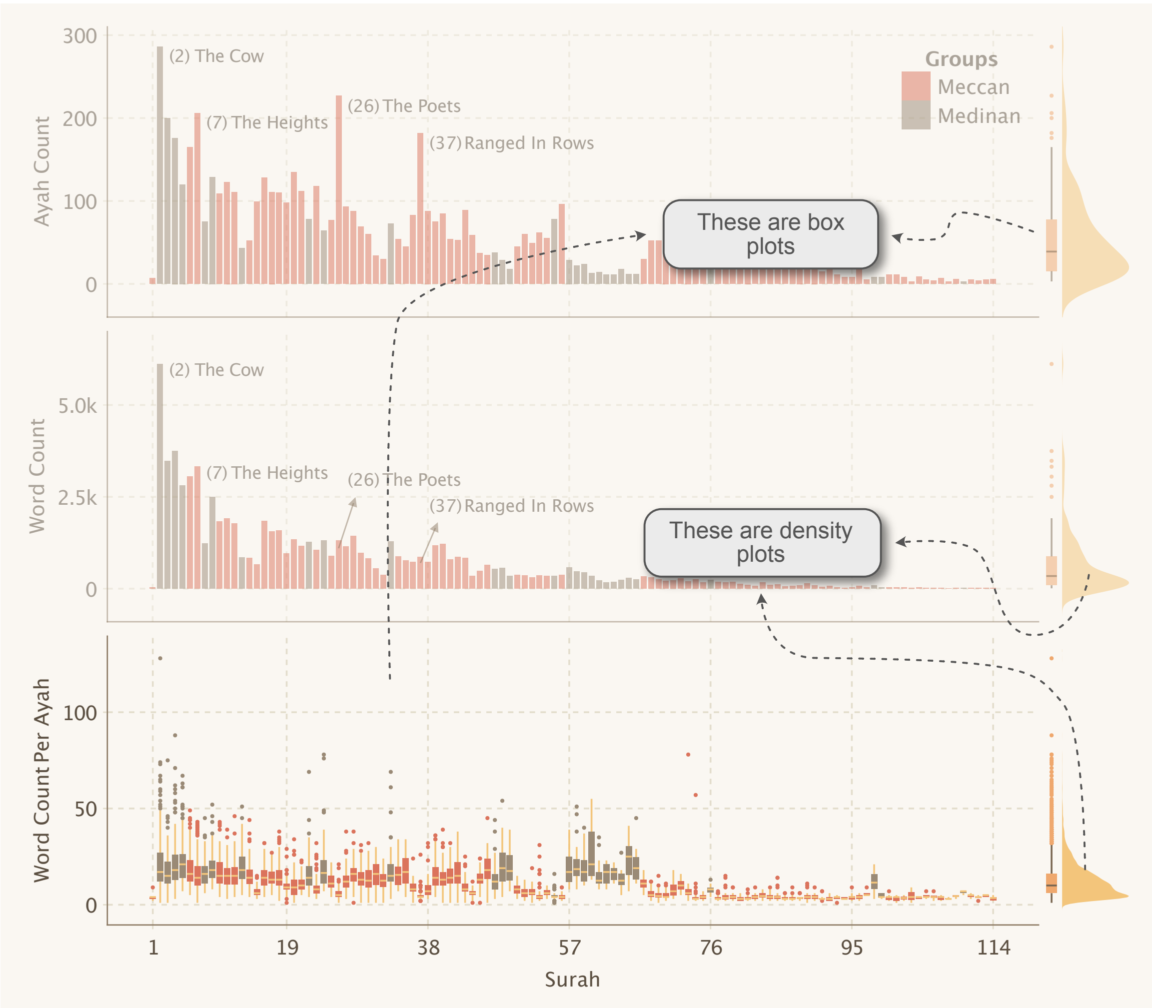


Figure 3.1: Statistics of the words and *ayāt* اٰيَات (verses) of the Qur'ān

sented in different style of visualization. In fact, Histogram is also used to describe the same information as the Box and Density, and the three are therefore related. So much so, that the three can be put into one graph. As to how to interpret these, readers are referred to for further details [to add reference].

Example 3.3.1 (Frequency Distribution)**.** Consider again the task of drawing an *ayāt* أَيَات from the Qur'ān, suppose the *ayāt* أَيَات are separated into مَكّيّة Meccan and مَدَنيّة Medinan, what is the probability of getting at most 10 *kalimāt* كَلِمَات or words in a sampled *ayāt* أَيَات from مَكّيّة Meccan *surahs* سُور?

Solution: To answer this, Figure 3.2 shows the *histograms* with the *box plots* and the *rainclouds* plots. The figure shows the frequency of the *kalimāt* كَلِمَات or words in a sampled *ayāt* أَيَات. This frequency describes the distribution of the *kalimāt* كَلِمَات. To interpret this, the Medinan مَدَنيّة histogram shows that most of the *ayāt* أَيَات have about 10 to 20 *kalimāt* كَلِمَات or words in total. This conclusion is based on the where the box of the box plot is located, which also corresponds to the area where the bars of the histogram are high, and also where most of the points or 'droplets' from the rainclouds plot are congested. With that said, *historgram*, *box plot*, and *rainclouds* are related and are telling the same story from different perspectives. It should be noted that, the rainclouds plot is not a common visualization tool.

Now, comparing the numbers from Medinan مَدَنيّة to the *āyāt l-makkiyyatu* آيَات المَكّيّةُ, there are about 5 to 15 *kalimāt* كَلِمَات to expect per *ayāt* أَيَات based on Figure 3.2.

The question has not been answered yet though, the above discussions only explains how to interpret the graphs in Figure 3.2. So to answer the question, one simply needs to total the number of *āyāt* آيَات with at most 10 *kalimāt* كَلِمَات or words and divide this with the total number of *āyāt l-makkiyyatu* آيَات المَكّيّةُ. The answer is as follows, and this is part of the result of this paper: there are 4613 *āyāt l-makkiyyatu* آيَات المَكّيّةُ, and out of these is 2602 *āyāt l-makkiyyatu* آيَات المَكّيّةُ with at most 10 words. Therefore, the probability is $\frac{2602}{4613} = 0.612$ or 61.2% probability. Formally, if X is the random variable of the event of observing at most 10 *kalimāt* كَلِمَات in a *āyāt l-*

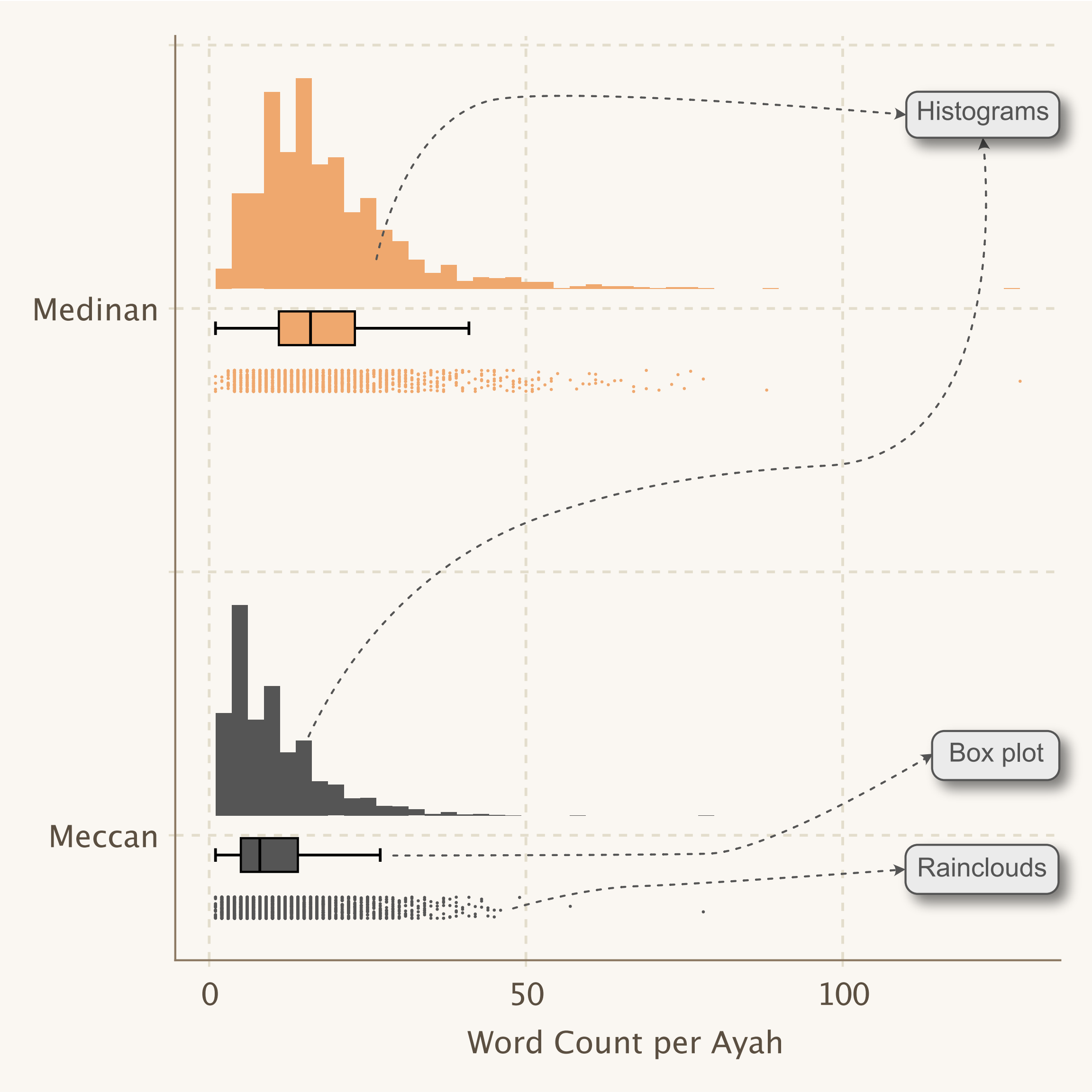


Figure 3.2: Probability density function plot of word count per *ayāt* أَيَات by revelation location, in relation to its box plot and rainclouds.

makkiyyatu آيَات المكِّيَّة, then

$$p(X \leq 10) = \sum_{x=0}^{10} p(X = x) \tag{3.16}$$

$$= p(X = 0) + p(X = 1) + p(X = 2) + \cdots + p(X = 10) \tag{3.17}$$

$$= 0 + \frac{24}{4613} + \frac{172}{4613} + \cdots + \frac{222}{4613} \tag{3.18}$$

$$= \frac{2602}{4613} = 0.612 \tag{3.19}$$

From Eq. 3.17, $p(X = 0)$ is the probability of observing zero *kalima* كَلِمَة in a *āyāt* *l-makkiyyatu* آيَات المكّيّةُ, and $p(X = 1)$ is the probability of observing one *kalima* كَلِمَة in a *āyāt l-makkiyyatu* آيَات المكّيّةُ, and so on. The numbers in Eq. 3.18 are the number of *āyāt l-makkiyyatu* آيَات المكّيّةُ having zero, one, two, to ten *kalimāt* كَلِمَات. □

3.4 Population and Sample

Statistics is a branch of science that is concerned with understanding how the data behave based on a sample—a small set of the said data. It uses statistical methodologies to understand these behavior such as probability mass/density function, and use the findings from these tools on the sampled data as a conclusion for the population—the overall data.

Figure 3.3 illustrates the relation of population and sample data. A good example of this is the political surveys on the pulse of the nation on the candidates prior to election. Private entities like PulseAsia[1] and Social Weather Station[2] (SWS) do their survey by sampling from the total population of the Philippines, hence the name survey.

The statistics computed from the surveys like the percentage of votes for particular political candidate are referred as estimates, this is because the computation was done in a sample of the population and not on the population itself. It is therefore important that for these estimates to be accurate representation of the nation's opinion, it has to be representative of the population. That is, the sample shouldn't be bias and leaning to the opinions of the few only and not of the whole nation.

The importance of the sample data as illustrated below follows from the fact that it saves time and cost since interviewing 2500 compared to 100,000 is better compromise for the estimated vote percentages. Further, since these samples requires to well represent the population, the computations of the statistics are estimated using statistical models, like probability mass/density functions, and other models like linear and nonlinears discussed in Section 3.6. The next example will illustrate the idea population and sample, and how probabilistic modeling can help in un-

[1] `https://pulseasia.ph/`

[2] `https://www.sws.org.ph/swsmain/home/`

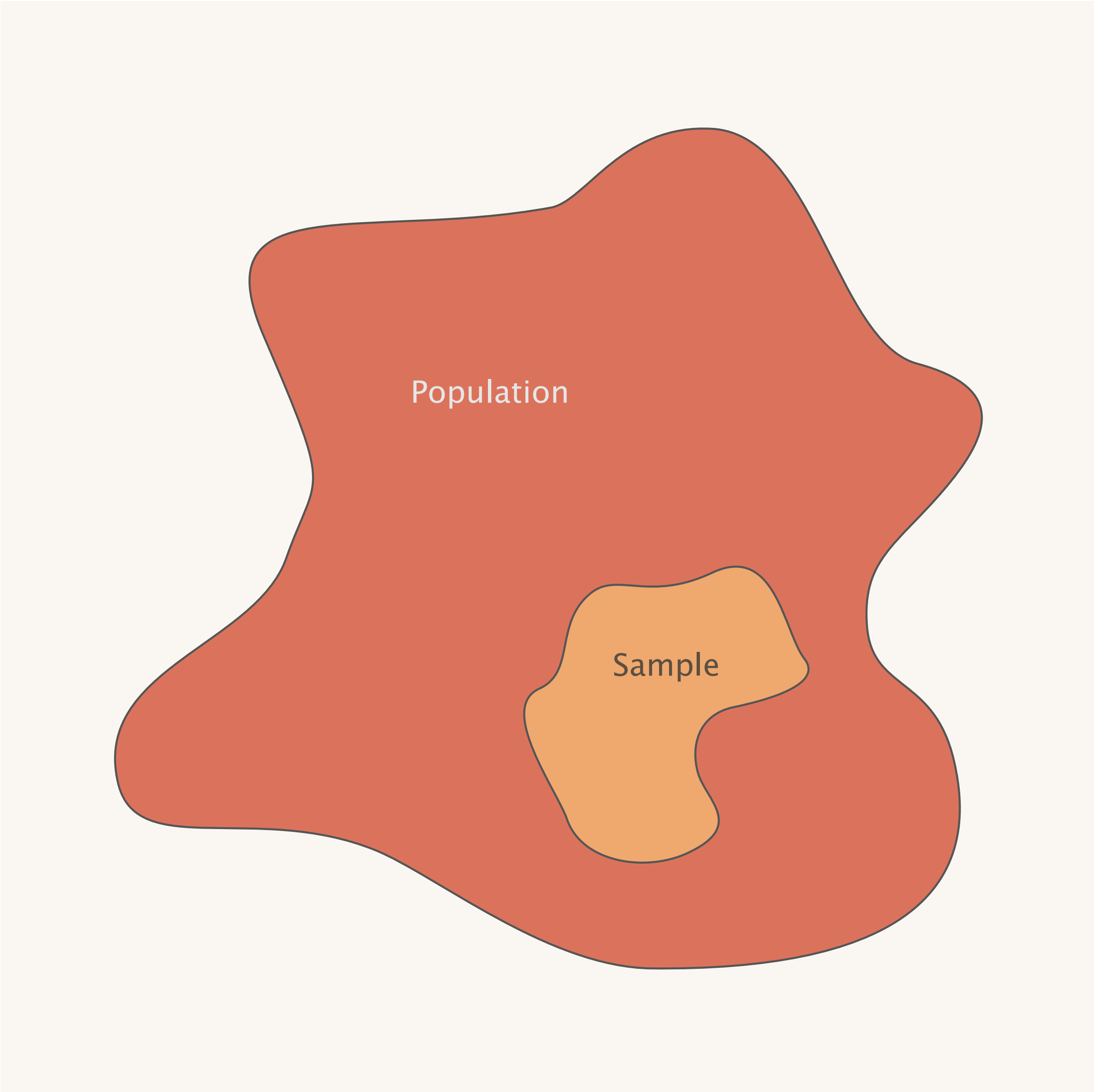


Figure 3.3: Population and sample illustration

derstanding insights and answering more questions.

Example 3.4.1. Consider the task of sampling 250 *āyāt* آيَات from the population of *āyāt l-makkiyya*, describe the statistics of the population and the sample.

Solution: In Statistics, there are several ways to sample from data, the simplest approach is through the use of *uniform distribution*, that is, the sampling assumes that all data from the population are distributed equally, in the sense that all data points have equal chance of being selected. The other approach is through a *weighted distribution*, where a probability is assigned to each of the data points in the population, so that, the selection will be biased to those with high probability. Figure 3.4 shows

Table 3.1: Descriptive statistics of the population and sample data of *kalimātu l-makkiyya* كَلِمَاتُ المَكِّيَّة

Data	Mean	Median	Variance	Std. Deviation
Population	10.28	8	54.15	7.36
Sample	10.64	9	48.96	7.00

the graphs or plots of the population distribution of *kalimātu l-makkiyya* كَلِمَاتُ المَكِّيَّة, which is presented as the top plot, this distribution of كَلِمَاتُ المَكِّيَّة is the same one shown in the bottom plot of Figure 3.2.

To draw 100 samples from the said population, a simple random sampling without replacement (SRSWOR) is used for selecting or drawing samples. SRSWOR works by randomly selecting sample from the population and then setting it aside as the first sample. The resulting 100 samples are plotted in the bottom plot of Figure 3.4.

As discussed above, the sample needs to be representative of the population. Figure 3.4 shows that the sampling distribution has more or less the same shape as the population. So that, the statistics are shown in Table 3.1. From the said table, it can be seen that in terms of centrality, the both data are almost the same, for example the mean is 12.42 for the population and 10.64 for the sample. The reason why the mean in the population is much higher is due to the outlier in the population, which is seen in the extent of the tail of the Kernel Density Estimate in Figure 3.4. In fact, this is seen in the Median in Table 3.1, where the estimate are almost the same. This is because the median is not affected by the outlier. However, the variance did suffer from the outlier in the population, with 50% reduction in the sample variance, 54.15 to 48.96. The sample will less likely get the outlier as the sample since the outlier is only one observation from the 4163 total Meccan surahs.

The estimates got from the sample ideally are taken from a probabilistic model fitted into the sample data. This computation will be discussed in the next example.

□

Example 3.4.2. Consider again Ex. 3.4.1, suppose the sample data is the only data

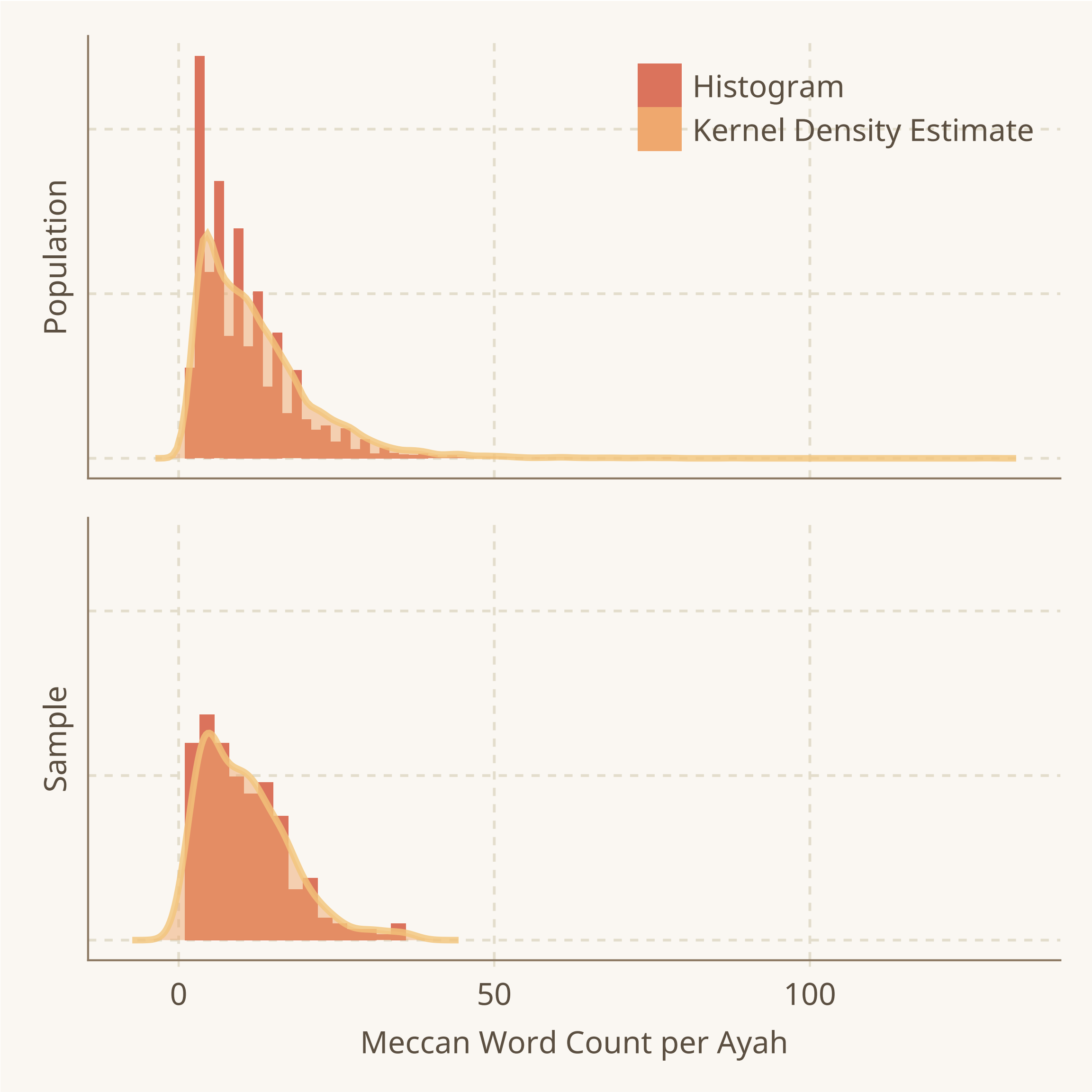


Figure 3.4: Population and sample distribution of Meccan

available, how will you compute the probability of getting exactly 35 *kalimātu l-sūratu l-makkiyyatu* كَلِمَاتُ السُّورَتُ المَكِّيَّةُ?

Solution: To answer this question, one might approach this using the frequency distribution as in Ex. 3.3.1. However, the use of frequency distribution from the said example is applicable since that deals with the population data, which is already the true probability, and there is no need to do some estimation. It is like try to get the average height of male Filipinos in the Philippines by doing census across the population, for such case, why would you do an estimate if you have all the census data of all heights of the Filipino, wouldn't it be easier to just do the average computation directly? This is the analogy for the frequency distribution used in Ex. 3.3.1, that is, no need to do some estimation. However, for this problem, the assumption is that only the sample is available, that is, not all of the population is available. With that said, the best solution is to do some estimation. Now, doing an estimation is possible using the samples only, that is, using the idea of frequency distribution but this time applying it on the sample. This is becaus the sample was done using a random sampling, which more or less representative of the population. However, using this approach may possess a problem, let's see what that problem will likely be by forcing the approach in Ex. 3.3.1, following the computation as in Eq. (3.16) to Eq. (3.19).

Let X again be the random variable of an event of observing 35 *kalimāt* كَلِمَات, this time from the sample, then

$$p(X = 35) = \begin{cases} \dfrac{0}{250} = 0, & \text{if sample data} \\ \dfrac{5}{4613} = 0.0011, & \text{if population data} \end{cases} \tag{3.20}$$

From the Eq. 3.20, it shows that if we use the sample data for estimating the probability of observing 35 كَلِمَات, then the answer above is 0, meaning by estimate there is a 0 chance of observing a 35 كَلِمَات from a *āyātu l-makkiyyatu* السُّورَاتُ المَكِّيَّةُ. This conclusion is indeed misleading, since according to the population data, there are *āyātu l-makkiyyatu* آيَاتُ المَكِّيَّةُ that has 35 *kalimāt* كَلِمَات.

So, how to properly estimate this then? This is where the concept of probabilistic modeling comes in. For this problem, the data is count, and that the event of observing 10 *kalimātu l-sūratu l-makkiyyatu* كَلِمَاتُ السُّورَتُ المَكِّيَّةُ in an أَيَة is known to be best modeled by a Poisson distribution defined in Defn. 3.5.1. Ex. 3.5.1 will discuss how to solve this.

3.5 Probability Distributions

Definition 3.5.1 (Poisson Mass Function)**.** Let X be a random variable and let $\lambda > 0$ be a parameter, then if x is the random value, then the *Poisson* mass function is given by:

$$p(X = x) = \frac{\lambda^x \exp(-\lambda)}{x!} \tag{3.21}$$

Example 3.5.1. Consider again Ex. 3.4.1, the problem can be solved using the Poisson distribution.

Definition 3.5.2 (Normal Density Function)**.** Let Y be a random variable and let $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}$ be the mean and variance parameters, if y is the random value, then the *Gaussian* or *Normal* density function is given below:

$$p(Y = y) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{\sigma^2}\right\}, \quad \text{where } -\infty < y < \infty \tag{3.22}$$

Definition 3.5.3 (Dirichlet Density Function)**.** Let $\mathbf{Y}$ be a vector random variable with a random value $\mathbf{y} := [y_1, \cdots, y_k]^{\mathrm{T}}$ and let $\boldsymbol{\alpha} := [\alpha_1, \cdots, \alpha_k]^{\mathrm{T}}$ be the parameters, then the *Dirichlet* density function is defined as

$$p(\mathbf{X} = \mathbf{x}; \boldsymbol{\alpha}) := \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1}, \tag{3.23}$$

where,

$$B(\boldsymbol{\alpha}) := \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)} \tag{3.24}$$

□

Definition 3.5.4 (Multinomial Mass Function)**.** Let $\mathbf{X}$ be a vector random variable with a random value $\mathbf{x} := [x_1, \cdots, x_k]^{\mathrm{T}}$ such that $x_i \geq 0, \forall i \in [1, k]$, let $\boldsymbol{q} := [q_1, \cdots, q_k]^{\mathrm{T}}$ and n be the parameters, the probability mass function of a Multinomial distribution is

$$f(x_1, \cdots, x_k; n, q_1, \cdots, q_k) := q(X_1 = x_1 \text{ and } \cdots \text{ and } X_k = x_k)$$
$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} q_1^{x_1} \times \cdots \times q_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0 & \text{otherwise,} \end{cases} \tag{3.25}$$

3.6 Statistical Modeling

Probability distributions are fundamental models that can be used to describe some characteristics of the data. However, combinations of variables, and accounting dynamics of the observations can lead into other types of models as well. In general, a statistical model can be written using the following formula:

$$y = f(x|\mathbf{\Theta}) + \varepsilon, \tag{3.26}$$

where $y \sim p(\cdot)$ and $\varepsilon \sim p(\cdot)$.

3.6.1 Frequentist Estimation

There are two main ways to estimating the parameters $\mathbf{\Theta}$, and that is either through Frequentist or Bayesian approach.

Maximum Likelihood Estimation

Numerical Approximation

3.6.2 Bayesian Estimation

Markov Chain Monte Carlo

Variational Bayes

Inverse Laplace Method

3.7 Types of Models

3.7.1 Parametric

3.7.2 Nonparametric

3.7.3 Supervised

3.7.4 Unsupervised

Chapter 4

Background on Neural Networks

Detailed presentation on the theoretical results of ANN will be the focus of this chapter. Starting with the discussion on different types of *neurons* and a brief history about it. In Section 4.3, these neurons are then arranged to form networks which will be the topology of feedforward *multilayer perceptron* (MLP). Following that, are definitions of the standard notations for ANN used for the rest of this thesis. Then what follows is the optimization on the parameters. And finally, the extension to Bayesian framework for ANN.

4.1 Perceptron

From ((?, ?)), some 15 years after the publication of ((?, ?)) classic paper, a new approach to the pattern recognition was introduced by (Rosenblatt 1958) in his work on the *perceptron*, a novel method of supervised learning. The crowning achievement's of Rosenblatt's work was the so-called *perceptron convergence theorem*, the first proof for which was outlined by (Rosenblatt 1960b).

Figure 4.1: Structure of a Typical Biological Neuron.[1]

Figure 4.1 is the typical nerve cell in Biology. In relation to perceptron, the dendrite in this case takes several binary inputs, and returns a single binary output on its axon. Mathematically, let $w_1, w_2, \cdots, w_p$ be the parameters or weights, and $x_1, x_2, \cdots, x_p$ be the inputs. Then the perceptron is defined as follows:

$$y = \begin{cases} 0, & \sum_i w_i x_i \leq c \\ 1, & \sum_i w_i x_i > c \end{cases} \tag{4.1}$$

where c is a threshold set by the experimenter. The following diagram in Figure 4.2 is the schematic representation of Equation (4.1).

[1] Original SVG image by Quasar Jarosz but colors and some labels were modified for this paper.

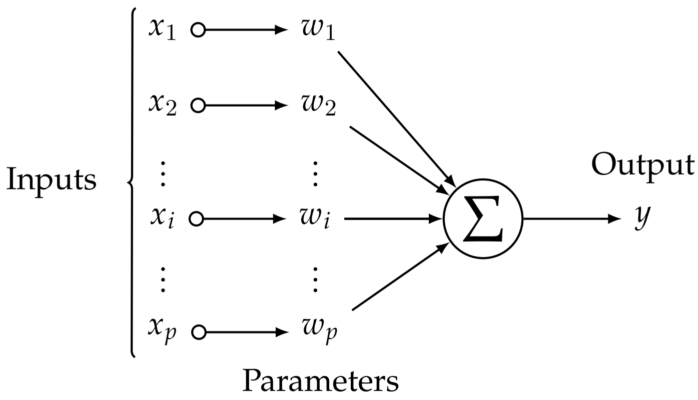


Figure 4.2: Schematic Representation of the Perceptron Neuron.

One limitation of the perceptron is its binary output, which sometimes is poor in discrimination problem, such as in imaging. To address this issue, a new artificial neuron is proposed that returns continuous values between 0 and 1, and so it can be interpreted as probabilities.

4.2 Logistic Sigmoid Neuron

The logistic sigmoid neuron or simply sigmoid neuron takes its name from the fact that it uses logistic function as its link/basis function, thus it is also known as logistic neuron. Formally, let w_0 be the intercept or the bias term and as before let $w_1, w_2, \cdots, w_p$ be the parameters, then the logistic sigmoid function is given by

$$z = \sum_i w_i x_i + w_0, \quad y = \varrho(z) = \frac{1}{1 + \exp(-z)} \tag{4.2}$$

The following diagram (Figure 4.3) depicts the flow of the computation in the sigmoid neuron. Also Figure 4.4 is the plot of the sigmoid function in a two dimensional space. In the proceeding section, the connections between neurons will be discussed, and how the architecture of these networks is designed.

4.3 Topology

The basic architecture of the network comprises of the *input* and the *output* layers, between them are the *hidden* layers. These hidden layers are layers that are not observed by the experimenter, hence the name. Figure 4.5 shows the diagram of a *feedforward* neural network. It is feedforward due to the fact that the inputs are

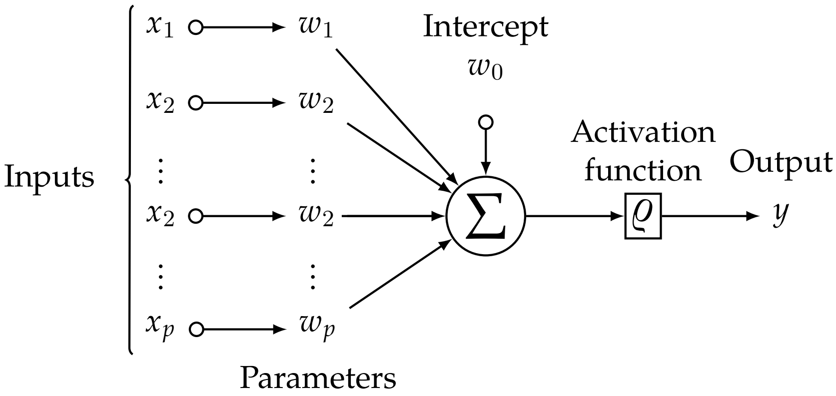


Figure 4.3: Diagram of the Logistic Neuron Computation.

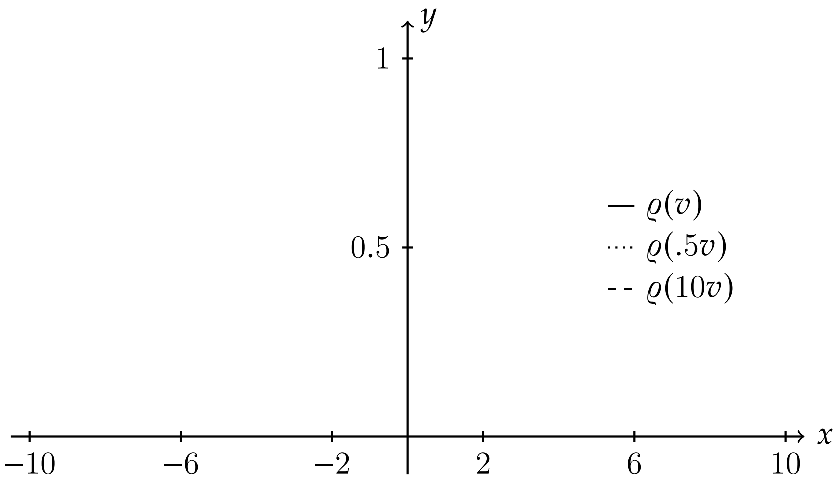


Figure 4.4: Plot of the Sigmoid Function.

propagated onto the succeeding layers up to the output without redirecting back to the previous layer after processing. Otherwise the model is called *recurrent* neural network.

Since the input and output layers are at best observed, the hidden layers will only serve as behind-the-scene computations. And the parameter associated with each neuron has no interpretation at all, and that's one major limitation of ANN. Neural networks can have several hidden layers depending on the study, for high level of abstraction one might consider multiple layers also known as *multilayer perceptrons* (MLP), which is a misnomer since layers are made up of logistic sigmoid neurons not of perceptrons. Networks with 2 or more hidden layers are often referred to as *deep neural networks* ((?, ?)), an example of this is depicted in Figure 4.6. The unknown vector $\mathbf{w} = [(w_i)]_{i=0}^{p}$ in Equation (4.2), is the parameter vec-

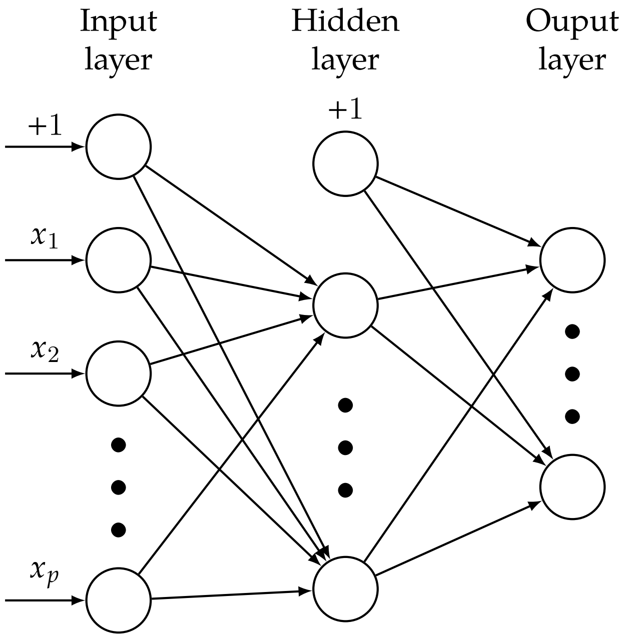


Figure 4.5: Feedforward Neural Network.

tor that needs to be estimated. And this is done by maximum likelihood which is equivalent to minimization of the error function through mathematical programming using methods such as *Gradient Descent* algorithm. The succeeding section will formalize the standard notations, and then the definition of the loss function which will be the basis for choosing the best estimates of the weights, and finally the estimation.

4.3.1 Notations

The following definition will serve as the standard notation for the rest of the succeeding chapters. To start with, let $f : \mathcal{X} \to \mathcal{Y}$ be the unknown *target function* of the data points in the population, that is $y = f(\mathbf{x})$. The sample data set from this population is denoted by $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$. The learning algorithm picks $g : \mathcal{X} \to \mathcal{Y}$ as the *final hypothesis* from the hypothesis set $\mathscr{H} = \{h_v : h_v(\mathbf{x}) = \hat{y}, v = 1, 2, \cdots, V\}$ such that $g \approx f$, that is, g is the hypothesis that best approximate the target function f, thus $\hat{y} = g(\mathbf{x})$. The error in the training data set named "in-sample error" is denoted by E_{in}, while the error in the validation data set named "out-of-sample error" is denoted by E_{out}.

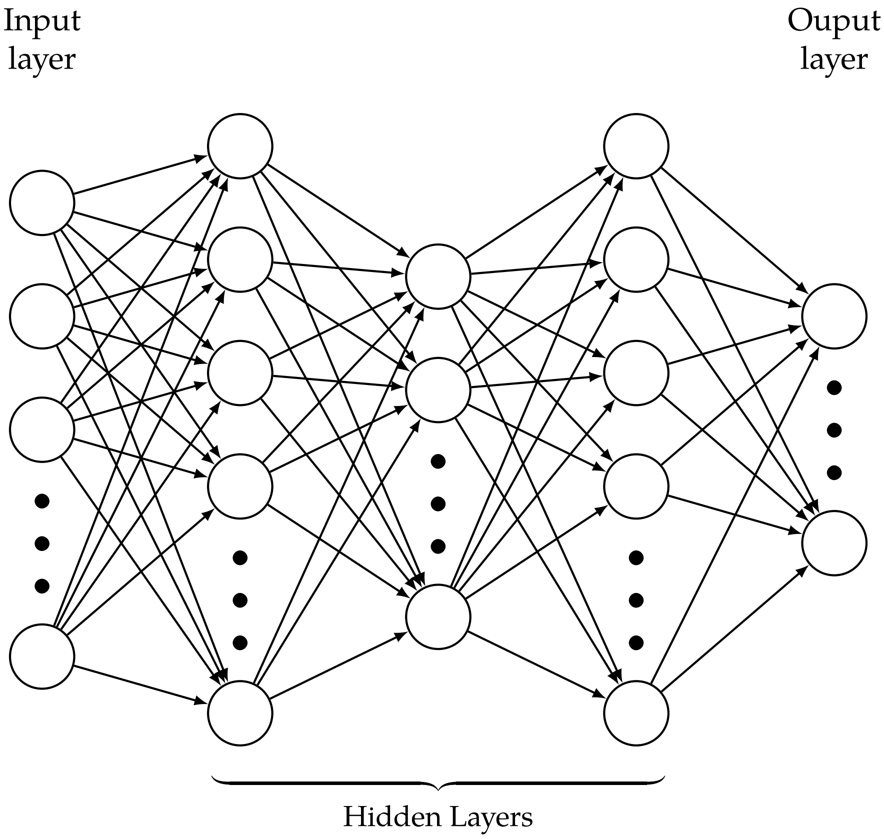


Figure 4.6: MULTILAYER FEEDFORWARD NEURAL NETWORK.

For ANN models, let i, j and l be the indices of the observations in the inputs, outputs and the layers containing the observations, respectively. The domain of these indices are as follows: $i \in [0, d^{(l-1)}]$, $j \in [1, d^{(l)}]$ and $l \in [1, L]$, where $d^{(l)}$ is the dimension of the outputs in the lth layer. The upper bound of the inputs is $d^{(l-1)}$, since this is the dimension of the outputs in the previous layer, $l-1$, and is now an input layer due to the foward-propagation process, in this case the next layer has $d^{(l)}$ dimension. Further, the lower bound of the input is 0 to account for the intercept term w_0. Hence the following is the standard notation of the parameters,

$$w_{ji}^{(l)} = \begin{cases} 1 \leq l \leq L & \text{layers} \\ 0 \leq i \leq d^{(l-1)} & \text{inputs} \\ 1 \leq j \leq d^{(l)} & \text{outputs} \end{cases}$$

Let $\mathbf{x}_k$ be the training input, $k = 1, 2, \cdots, K$. Since the input layer is the 0th layer with d dimension, then each input is a $d^{(0)} \times 1$ vector. That is,

$$\mathbf{x}_k = \begin{bmatrix} x_{ki}^{(0)} \\ x_{k2}^{(0)} \\ \vdots \\ x_{kd^{(0)}}^{(0)} \end{bmatrix}$$

For example in imaging, each entry in the vector corresponds to the gradient of grayscale pixel. In this setting, $h(\mathbf{x}_k, \mathbf{w})$ denotes the hypothesis or the model, where $\mathbf{w}$ is the vector of all weights. So different network achitecture relates to different model, and the final hypothesis is denoted by $g(\mathbf{x}_k, \mathbf{w})$.

The feedforward process executes as follows: the vector $\mathbf{x}_k$ will serve as an input on the first hidden layer, whose output will then be the input on the next hidden layer, and so on until the output layer. In general,

$$x_{kj}^{(l)} = \varrho\left(z_{kj}^{(l)}\right) = \varrho\left(\left\{\mathbf{w}_j^{(l)}\right\}^{\mathrm{T}} \mathbf{x}_k^{(l-1)} + w_{j0}^{(l)}\right) = \varrho\left(\sum_{i=1}^{d^{(l-1)}} w_{ji}^{(l)} x_{ki}^{(l-1)} + w_{j0}^{(l)}\right),$$

where $\mathbf{w}_j^{(l)} = \begin{bmatrix} w_{j1}^{(1)} & w_{j2}^{(2)} & \cdots & w_{jd^{(l-1)}}^{(l)} \end{bmatrix}^{\mathrm{T}}$. This is done recursively, that is the output $x_{kj}^{(l)}$ will be the input on the next layer. For example, the following is the formula for obtaining $x_{kj}^{(l+1)}$,

$$\begin{aligned} x_{kj}^{(l+1)} &= \varrho\left(z_{kj}^{(l+1)}\right) = \varrho\left(\left\{\mathbf{w}_j^{(l+1)}\right\}^{\mathrm{T}} \mathbf{x}_k^{(l)} + w_{j0}^{(l)}\right) = \varrho\left(\sum_{i=1}^{d^{(l)}} w_{ji}^{(l+1)} x_{ki}^{(l)} + w_{j0}^{(l+1)}\right) \\ &= \varrho\left\{\sum_{i=1}^{d^{(l)}} w_{ji}^{(l+1)} \left[\varrho\left(\sum_{i=1}^{d^{(l-1)}} w_{ji}^{(l)} x_{ki}^{(l-1)} + w_{j0}^{(l)}\right)\right] + w_{j0}^{(l+1)}\right\}. \end{aligned}$$

So the dimension d will vary from layer to layer. Consider Figure 4.5, the neurons can now be labelled with the standard notations as in Figure 4.7. So that $x_{kj}^{(1)}$ in the hidden layer has the following form:

$$x_{kj}^{(1)} = \varrho(z_{kj}^{(1)}) = \varrho\left(\left\{\mathbf{w}_j^{(1)}\right\}^{\mathrm{T}} \mathbf{x}_k^{(0)} + w_{j0}^{(1)}\right) = \varrho\left(\sum_{i=1}^{d^{(0)}} w_{ji}^{(1)} x_{ki}^{(0)} + w_{j0}^{(1)}\right),$$

The $x_{kj}^{(2)}$ in the output layer is computed from the following equation,

$$\begin{aligned} x_{kj}^{(2)} &= \varrho(z_{kj}^{(2)}) = \varrho\left(\left\{\mathbf{w}_j^{(2)}\right\}^{\mathrm{T}} \mathbf{x}_k^{(1)} + w_{j0}^{(2)}\right) = \varrho\left(\sum_{i=1}^{d^{(1)}} w_{ji}^{(2)} x_{ki}^{(1)} + w_{j0}^{(2)}\right) \\ &= \varrho\left\{\sum_{i=1}^{d^{(1)}} w_{ji}^{(2)} \left[\varrho\left(\sum_{i=1}^{d^{(0)}} w_{ji}^{(1)} x_{ki}^{(0)} + w_{j0}^{(1)}\right)\right] + w_{j0}^{(2)}\right\}. \end{aligned}$$

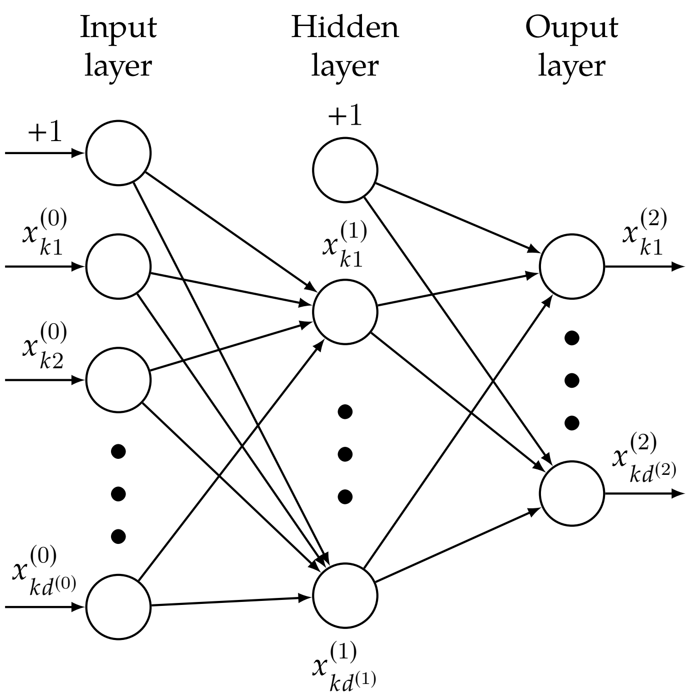


Figure 4.7: Neural Net with Standard Neuron Notation.

The final output $\mathbf{x}_k^{(2)}$ is the value of the hypothesis, that is $h(\mathbf{x}_k, \mathbf{w}) = \mathbf{x}_k^{(L)}$, where $L = 2$.

4.4 Network Training

Network parameter estimation is done through optimization, this is because ANN loss function has no unique stationary point. That is, it consist of local minima and

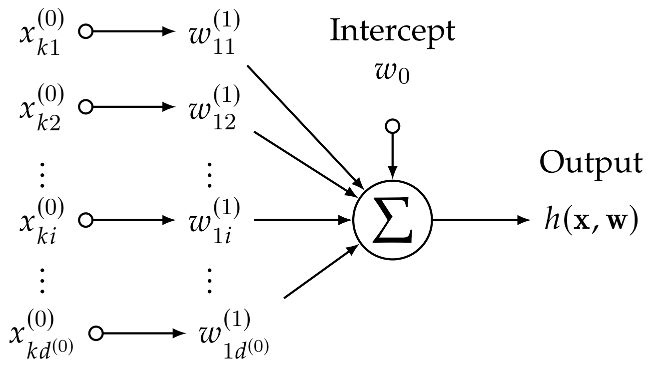


Figure 4.8: Neuron's Schematic Diagram for Linear Regression.

often one unique global minimum.

4.4.1 Loss Function

In this section, the error functions of ANN models for regression and classification are derived from probabilistic perspective, this will be useful when extending the model to full Bayesian treatment. For regression setup, the network has at least two nodes (one for the intercept and one for the independent variable) at the input layer and one node at the output layer. The standard activation function is the identity function, i.e. $h(\mathbf{x}_k, \mathbf{w}) = z_k = \mathbf{w}^{\mathrm{T}}\mathbf{x}_k$, so there is no hidden layer, see Figure 4.8. Given the training input $\mathbf{X} = [\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \cdots, \mathbf{x}_N^{(0)}]$, where $\mathbf{x}_k^{(0)}$ is a $d^{(0)}$-dimensional feature vector whose elements are $x_{k0}^{(0)}, x_{k1}^{(0)}, \cdots, x_{kd^{(0)}}^{(0)}$ $\forall k = 1, 2, \cdots, K$, the corresponding target variable is $\mathbf{Y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$, and the error term ε_k is assumed to be normally distributed with mean $\mathbb{E}\varepsilon_k = 0$ and variance $\mathbb{V}\text{ar}\varepsilon_k = \sigma^2$. It follows that the uncertainty around the target variable is also Gaussian distributed and is given by

$$\begin{aligned}\mathbb{P}(y|\mathbf{x}, \mathbf{w}, \sigma^2) &= \mathcal{N}(y|h(\mathbf{x}, \mathbf{w}), \sigma^2)\\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y - h(\mathbf{x}, \mathbf{w}))^2}{2\sigma^2}\right]\end{aligned} \tag{4.3}$$

If the data are independent and identically distributed, then the likelihood function of $\mathbf{y}$ is given by,

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \sigma|\mathbf{y}, \mathbf{X}) = \mathbb{P}(\mathbf{y}|\mathbf{X}, \mathbf{w}, \sigma) &= \prod_{k=1}^{K} \mathcal{N}(y_k|h(\mathbf{x}_k, \mathbf{w}), \sigma^2) \\ &= \frac{1}{(2\pi)^{K/2}\sigma^K} \exp\left[-\frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_k - h(\mathbf{x}_k, \mathbf{w}))^2\right]. \end{aligned} \tag{4.4}$$

Taking the log-likelihood,

$$\ell(\mathbf{w}, \sigma|\mathbf{y}, \mathbf{X}) = -\frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_k - h(\mathbf{x}_k, \mathbf{w}))^2 - K\log\sigma - \frac{K}{2}\log 2\pi. \tag{4.5}$$

The estimate for the weight $\mathbf{w}$, denoted by $\hat{\mathbf{w}}_{\text{MLE}}$, is obtained by maximizing the log-likelihood function. To do so, partial differentiation is applied to Equation (4.5) with respect to $\mathbf{w}$, suggesting that the last two terms in the right hand side can be disregarded since these are not dependent on $\mathbf{w}$. Also, the location of the maximum log-likelihood with respect to $\mathbf{w}$ is not affected by arbitrary positive scalar multiplication, so the factor $\frac{1}{\sigma^2}$ can be omitted. Equivalently, one can minimize the negative log-likelihood instead of maximizing ℓ. The reason for doing this has something to do with numerical computation, since optimizers in most statistical packages often work by minimizing the function. Thus differentiating negative of Equation (4.5) is the same as differentiating the following equation,

$$E_{\text{in}}(h) = \frac{1}{2}\sum_{k=1}^{K}(y_k - h(\mathbf{x}_k, \mathbf{w}))^2, \tag{4.6}$$

which is similar to the form of *residual sum-of-squares*. Hence the sum-of-squares error function is a consequence of maximum likelihood under the normality assumption of the uncertainty around the target variable.

Likewise, the precision parameter can be estimated using maximum likelihood,

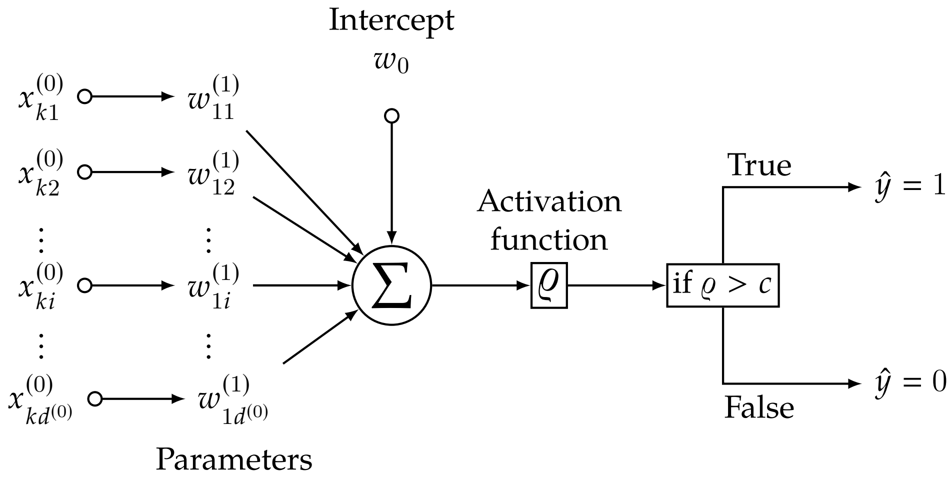


Figure 4.9: Neuron's Schematic Diagram for Logistic Regression.

by simply differentiating Equation (4.5) with respect to σ, i.e.

$$\hat{\sigma}^2_{\text{mle}} = \frac{1}{K}\sum_{k=1}^{K}(y_k - h(\mathbf{x}_k, \mathbf{w}))^2.$$

Now consider the case for classification, in which there is a single target variable y that returns binary output, cases like when $y = 1$ corresponds to class C_1 and $y = 0$ for class C_2. The appropriate activation function is the logistic sigmoid defined in Section 4.2 that is

$$\varrho(z) = \frac{1}{1 + \exp(-z_k)},$$

implying $\varrho : \mathbb{R} \to [0, 1]$, and hence it can be interpreted as probability, this type of model has the same network architecture and neuron's schematic diagram as that in linear regression model (see Figure 4.8), but this time the activation function has unit interval range with hypothesis given by

$$\hat{y} = h(\mathbf{x}, \mathbf{w}) = \begin{cases} 1, & \varrho > c \\ 0, & \varrho \leq c \end{cases},$$

where c is specified by the user, and is often chosen to be 0.5.

The probability of success, in this case $y = 1$ for C_1 is

$$\mathbb{P}(y = 1|h(\mathbf{x}, \mathbf{w})) = h(\mathbf{x}, \mathbf{w})$$

So that the conditional distribution of the target variable y given $\mathbf{x}$ is a Bernoulli distribution with density given by

$$\mathbb{P}(y|h(\mathbf{x}, \mathbf{w})) = \mathcal{B}(y|h(\mathbf{x}, \mathbf{w})) = h(\mathbf{x}, \mathbf{w})^y(1 - h(\mathbf{x}, \mathbf{w}))^{1-y}, \quad y \in \{0, 1\}.$$

So for sample data set $\{\mathbf{x}_k, y_k\}$ where $\mathbf{x}_k$ is a $d^{(0)}$-dimensional feature space and $y_k \in \{0, 1\}$ $\forall k = 1, \cdots, K$, if the ys are independent and identically distributed then the likelihood function is given by

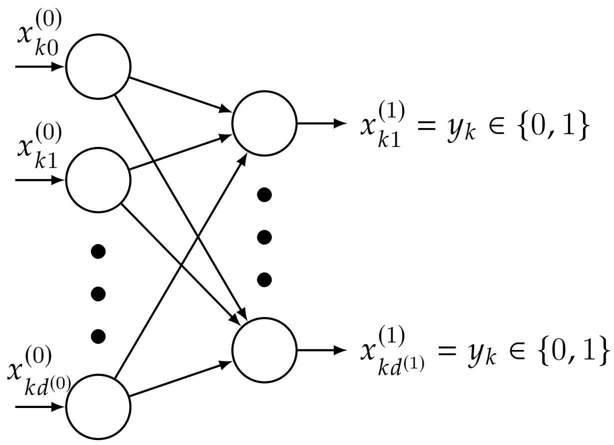
$$\mathcal{L}(h(\mathbf{x}_k, \mathbf{w})|\mathbf{y}) = \prod_{k=1}^{K} h(\mathbf{x}_k, \mathbf{w})^{y_k}(1 - h(\mathbf{x}_k, \mathbf{w}))^{1-y_k},$$

and the negative log-likelihood is given by

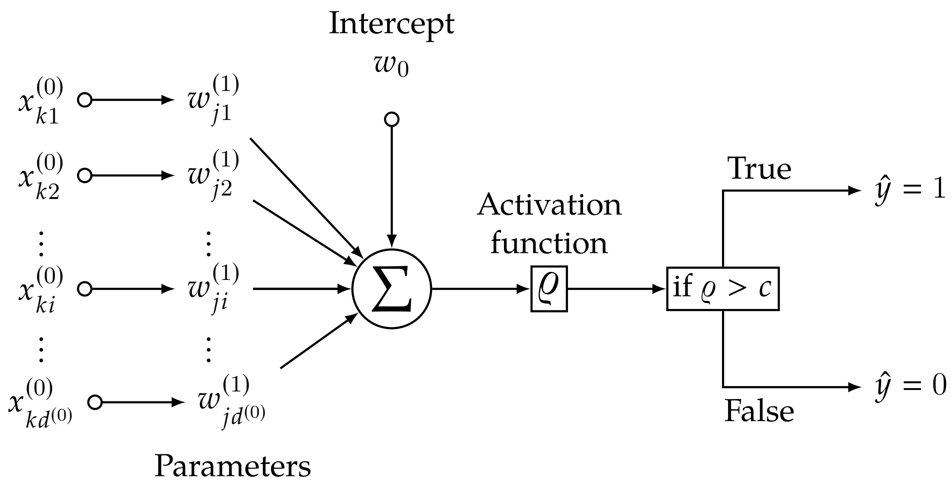
$$\begin{aligned} E_{\text{in}}(h) &= -\ell(h(\mathbf{x}_k, \mathbf{w})|\mathbf{y}) \\ &= -\sum_{k=1}^{K} \left[y_k \log h(\mathbf{x}_k, \mathbf{w}) + (1 - y_k) \log(1 - h(\mathbf{x}_k, \mathbf{w}))\right]. \end{aligned} \tag{4.7}$$

This equation is known as *cross-entropy* loss function. Extending the case to at least two separate binary classification requires $d^{(L)} \geq 2$ neurons on the output layer, each of which has a logistic sigmoid activation function. So the target variable is defined as $\mathbf{y}_k = [y_1, y_2, \cdots, y_{d^{(L)}}]^{\mathrm{T}}$, where $y_j \in \{0, 1\}$ $\forall j = 1, 2, \cdots, d^{(L)}$. Assuming independence on the class labels, the joint conditional distribution of the targets given the input vector is

$$\mathbb{P}(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \prod_{j=1}^{d^{(L)}} h(\mathbf{x}, \mathbf{w})^{y_j} \left[1 - h(\mathbf{x}, \mathbf{w})\right]^{1-y_j}.$$



(a) Network Architecture.



(b) Neuron's Schematic Diagram.

Figure 4.10: NEURAL NETWORK FOR $d^{(1)} \geq 2$ BINARY CLASSIFIER.

Then the likelihood function is given by

$$\mathcal{L}(h(\mathbf{x}_k, \mathbf{w})|\mathbf{y}) = \prod_{k=1}^{K} \prod_{j=1}^{d^{(L)}} h(\mathbf{x}_k, \mathbf{w})^{y_{kj}} \left[1 - h(\mathbf{x}_k, \mathbf{w})\right]^{1-y_{kj}} .$$

The negative log-likelihood then becomes,

$$\begin{aligned} E_{\text{in}}(h) &= -\ell(h(\mathbf{x}_k, \mathbf{w})|\mathbf{y}) \\ &= -\sum_{k=1}^{K} \sum_{j=1}^{d^{(L)}} \left[y_{kj} \log h(\mathbf{x}_k, \mathbf{w}) + (1 - y_{kj}) \log(1 - h(\mathbf{x}_k, \mathbf{w})) \right] . \end{aligned} \tag{4.8}$$

In general, the loss function of a neural network can be written as the average errors of the errors in the output layer across all k training inputs, i.e.

$$E_{\text{in}}(h) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{2}\sum_{j=1}^{d^{(L)}}(y_{kj} - h(\mathbf{x}_k, \mathbf{w}))^2 \tag{4.9}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\varepsilon(h(\mathbf{x}_k, \mathbf{w}), y_{kj}) \tag{4.10}$$

This equation is also known as the *mean squared error*. In order to minimize this function, one has to consider numerical methods such as *gradient descent* and *stochastic gradient descent*.

4.4.2 Back-propagation Algorithm

The average error of the k training inputs defined in Section 4.4.1 is given by

$$E_{\text{in}}(\mathbf{w}) \triangleq \frac{1}{K}\sum_{k=1}^{K}\varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k). \tag{4.11}$$

The terms in this equation are errors of the training inputs. That is, for each training input there is a corresponding error that needs to be minimized. This error is the difference between the target $\mathbf{y}_k$ and the output of the hypothesis $\mathbf{h}(\mathbf{x}_k, \mathbf{w})$. To account for the iteration counter r, let $\hat{\mathbf{w}}_r$ be the notation for the estimate of $\mathbf{w}$ at the rth iteration; also let ε_r be the error at the . The inner term of Equation (4.11) can be expanded as follows:

$$\begin{aligned}
\varepsilon(\mathbf{h}(\mathbf{x}_k, \hat{\mathbf{w}}_r), \mathbf{y}_k) &\triangleq \frac{1}{2}\|\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k, \hat{\mathbf{w}}_r)\|^2 = \frac{1}{2}\sum_{j=1}^{d^{(L)}}(y_{kj} - h_j(\mathbf{x}_k, \mathbf{w}))^2 \\
&= \frac{1}{2}\sum_{j=1}^{d^{(L)}}\left(y_{kj} - x_{kj}^{(L)}\right)^2 = \frac{1}{2}\sum_{j=1}^{d^{(L)}}\left[y_{kj} - \varrho\left(z_{kj}^{(L)}\right)\right]^2 \\
&= \frac{1}{2}\sum_{j=1}^{d^{(L)}}\left[y_{kj} - \varrho\left(\sum_{i=1}^{d^{(L-1)}} w_{ji}^{(L)} x_{ki}^{(L-1)} + w_{j0}^{(L)}\right)\right]^2.
\end{aligned} \tag{4.12}$$

For each training input k, gradient descent algorithm is applied to the function $\varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)$. The average of all this function correspond to the loss function, so the minimum average of the error terms will give the corresponding estimate of the parameters. For simplicity, assuming there is only one hidden layer, that is $L = 2$. Consider the following:

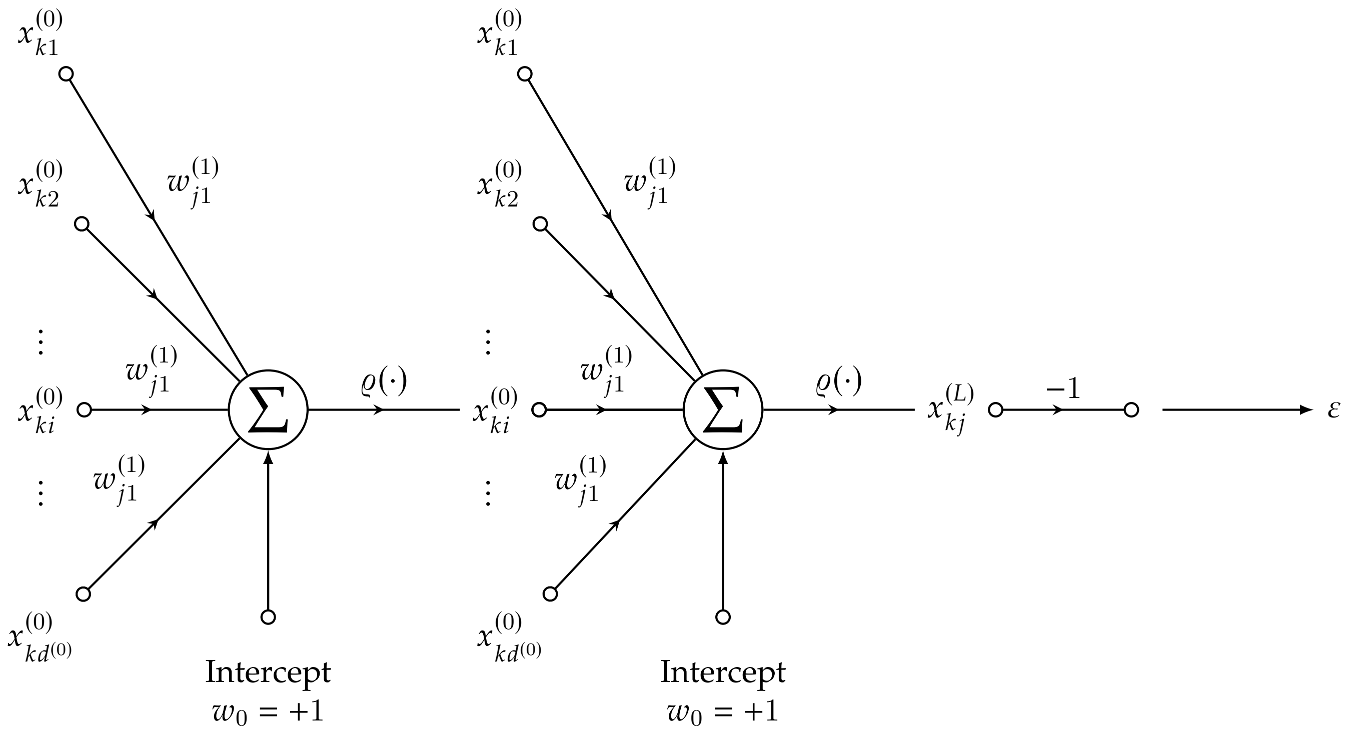


Figure 4.11: Signal-Flow Graph of Neurons (UNDERCONSTRUCTION).

$$\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial w_{ji}^{(L)}} = \frac{\partial E_{\text{in}}(\mathbf{w})}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)}{\partial w_{ji}^{(L)}} \tag{4.13}$$

where,

$$\begin{aligned}
\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} &= \frac{\partial}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{1}{K} \sum_{k=1}^{K} \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k) \\
&= \frac{1}{K} \sum_{k=1}^{K} \frac{\partial}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{1}{2} \sum_{j=1}^{d^{(L)}} (y_{kj} - h_j(\mathbf{x}_k, \mathbf{w}))^2 \\
&= \frac{1}{K} \sum_{k=1}^{K} \frac{\partial}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{1}{2} \sum_{j=1}^{d^{(L)}} \left[y_{kj} - x_{kj}^{(L)} \right]^2 \\
&= \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{d^{(L)}} \left[y_{kj} - x_{kj}^{(L)} \right],
\end{aligned} \tag{4.14}$$

and

$$\begin{aligned}
\frac{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)}{\partial w_{ji}^{(L)}} &= \frac{\partial}{\partial w_{ji}^{(L)}} \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{d^{(L)}} \left[y_{kj} - x_{kj}^{(L)} \right] \\
&= \frac{1}{K} \sum_{k=1}^{K} \left[-\left(y_{kj} - x_{kj}^{(L)} \right) \frac{\partial x_{kj}^{(L)}}{\partial w_{ji}^{(L)}} \right].
\end{aligned} \tag{4.15}$$

And note that $x_{kj}^{(L)} = \varrho\left(z_{kj}^{(L)}\right) = \varrho\left(\sum_{i=1}^{d^{(L-1)}} w_{ji}^{(L)} x_{ki}^{(L-1)} + w_{j0}^{(L)}\right)$. Hence,

$$\begin{aligned}
\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial w_{ji}^{(L)}} &= -\frac{1}{K} \sum_{k=1}^{K} \left(y_{kj} - x_{kj}^{(L)} \right) \frac{\partial x_{kj}^{(L)}}{\partial z_{kj}^{(L)}} \frac{\partial z_{kj}^{(L)}}{\partial w_{ji}^{(L)}} \\
&= -\frac{1}{K} \sum_{k=1}^{K} \left(y_{kj} - x_{kj}^{(L)} \right) \varrho'\left(z_{kj}^{(L)} \right) x_{ki}^{(L-1)}.
\end{aligned} \tag{4.16}$$

A gradient descent update at the $(r+1)$st iteration has the form,

$$w_{ji,r+1}^{(L)} = w_{ji,r}^{(L)} + \Delta w_{ji,r}^{(L)} = w_{ji,r}^{(L)} - \gamma \frac{\partial E_{\text{in}}(h)}{\partial w_{ji,r}^{(L)}}. \tag{4.17}$$

The $\Delta w_{ji,r}^{(L)}$ can be expressed as follows:

$$\Delta w_{ji,r}^{(L)} = -\gamma \frac{\partial E_{\text{in}}(h)}{\partial w_{ji,r}^{(L)}} = \gamma \delta_j^{(L)} x_{ki}^{(L-1)}, \tag{4.18}$$

where $\delta_j^{(L)}$ is the *local gradient* defined as

$$\begin{aligned}\delta_j^{(L)} &= -\frac{\partial E_{\text{in}}(h)}{\partial z_{kj}^{(L)}} = -\frac{\partial E_{\text{in}}(h)}{\partial x_{kj}^{(L)}} \frac{\partial x_{kj}^{(L)}}{\partial z_{kj}^{(L)}} = \frac{1}{K}\sum_{k=1}^{K} \varepsilon(h_j(\mathbf{x}_k, \mathbf{w}), y_{kj}) \varrho'\left(z_{kj}^{(L)}\right) \\ &= \frac{1}{K}\sum_{k=1}^{K} \left(y_{kj} - x_{kj}^{(L)}\right) \varrho'\left(z_{kj}^{(L)}\right). \end{aligned} \tag{4.19}$$

Thus the local gradient is the product of the error signal at the jth neuron of the Lth layer and the derivative of the associated activation function.

For weights at $(L-1)$th layer, the local gradient is then given by

$$\delta_j^{(L-1)} = -\frac{\partial E_{\text{in}}(h)}{\partial x_{kj}^{(L-1)}} \frac{\partial x_{kj}^{(L-1)}}{\partial z_{kj}^{(L-1)}} = -\frac{\partial E_{\text{in}}(h)}{\partial x_{kj}^{(L-1)}} \varrho'\left(z_{kj}^{(L-1)}\right). \tag{4.20}$$

where,

$$\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial x_{kj}^{(L-1)}} = \frac{\partial E_{\text{in}}(\mathbf{w})}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)}{\partial x_{kj}^{(L)}} \frac{\partial x_{kj}^{(L)}}{\partial z_{kj}^{(L)}} \frac{\partial z_{kj}^{(L)}}{\partial x_{kj}^{(L-1)}}. \tag{4.21}$$

And since

$$\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)} \frac{\partial \varepsilon(\mathbf{h}(\mathbf{x}_k, \mathbf{w}), \mathbf{y}_k)}{\partial x_{kj}^{(L)}} = -\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{d^{(L)}} \left(y_{kj} - x_{kj}^{(L)}\right) \tag{4.22}$$

then

$$\begin{aligned}
\frac{\partial E_{\text{in}}(\mathbf{w})}{\partial x_{kj}^{(L-1)}} &= -\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{d^{(L)}}\left(y_{kj} - x_{kj}^{(L)}\right)\frac{\partial x_{kj}^{(L)}}{\partial z_{kj}^{(L)}}\frac{\partial z_{kj}^{(L)}}{\partial x_{kj}^{(L-1)}} \\
&= -\sum_{j=1}^{d^{(L)}}\frac{1}{K}\sum_{k=1}^{K}\left(y_{kj} - x_{kj}^{(L)}\right)\varrho'\left(z_{kj}^{(L)}\right)w_{ji}^{(L)} \\
&= -\sum_{j=1}^{d^{(L)}}\delta_j^{(L)}w_{ji}^{(L)}. \qquad (4.23)
\end{aligned}$$

So that using Equation (4.23) in (4.20),

$$\delta_j^{(L-1)} = \varrho'\left(z_{kj}^{(L-1)}\right)\sum_{j=1}^{d^{(L)}}\delta_j^{(L)}w_{ji}^{(L)}. \qquad (4.24)$$

And just to emphasize $\varrho'\left(z_{kj}^{(L-1)}\right)$ is indexed at j but at the $(L-1)$th layer, so the dimension on that layer is not necessarily the same with the Lth layer, and thus it is not included in the summation. Equations (4.24) is known as the *back-propagation equations*.

4.5 Transformers

Transformer models is a model based on the work of (Vaswani et al., 2017).

Chapter 5

Background on Natural Language Processing

This chapter will discuss the concept and examples of Natural Language Processing in the context of analyzing the Qur'ān.

5.1 Text Analytics

Technically, any information can be regarded as data, whether it be image, audio, video, and even texts; and in fact many more. All of these raw data, however, needs to be translated into numbers, and therefore true for texts as well. There are several ways to translate texts into numbers, the easiest way maybe is to map the letters into numbers. Like for example, a will be 1, b will be 2, and so on. This simple assignment, while easy to do, disregards a lot of information or representation of the texts. Therefore, when it comes to mapping raw data not in numeric form, the assignment has to be logical. One needs to make sure that the transformation from its raw form needs to account the characteristics of the texts. For example, any language has words that have synonyms and antonyms. So, mapping these to numbers should account these relations as well. For example, since synonyms depict the related words to a given words, their numeric form should therefore be close to each other. So that, if "beautiful" corresponds to 132, then "attractive" should have a numeric mapping that is close to 132 since the word "beautiful" and "attractive" are considered synonymous or related. On the other hand, "ugly", which is an antonym of the "beautiful", the corresponding numeric form should be far from 132 to capture the opposite of the two in number. The next section will discuss formal mathematical definition for this concept of representing the texts.

5.2 Tokenization

5.3 Embeddings

Embeddings is the technical word for translating the words into numbers, it is the process of 'embedding' the words in a Euclidean vector space, a low-dimensional dense space. Let's understand this piece by piece. A logical approach to translating a word into numbers would be to do a one-hot encoding, which is a process of representing a word in a sparse vector with only one non-zero entry. Let's say the language contains only three words, 'hi', 'hello', and 'yes'. Then, the one-hot encoding for this would be:

$$\text{hi} = [1, 0, 0] \tag{5.1}$$

$$\text{hello} = [0, 1, 0] \tag{5.2}$$

$$\text{yes} = [0, 0, 1] \tag{5.3}$$

Notice, that each of the words has unique identity in their vector encoding. However, this example only shows 3 words, what if the language contains one million words, then each words will have 999,999 zeros in their vector encoding and only one non-zero entry. If a vector contains many zeros than non-zeros, then it is refered to as sparse. For this case, it is very sparse indeed. Further, applying such methodology is not flexible, since it is tied to fix number of words, which is a problem even for Arabic due to its complex morphology. Further, sparse vectors are not good for natural language processing (NLP), especially for doing all of the linear algebra computations. It is more preferrable to have a dense vector that also captures the semantics of the words.

Ideally, this very high dimensional vectors should be condensed into very low dimension to avoid the curse of dimensionality, a case where the vectors which supposed to be close to each other, but becomes more unique due to very high dimension. Apart from this, the embedding in the vector space should also capture the semantic proximity, that is, words that are similar semantically should be closed to each other in the embedding space. There are two approaches: *Term Frequency - Inverse Document Frequency*, and *Word Embeddings*.

5.3.1 Term Frequency - Inverse Document Frequency

The TF-IDF or Term Frequency - Inverse Document Frequency is a NLP method for embedding words. It is composed of two statistics, the TF and the IDF. The following is the formal definitions of both statistics.

Definition 5.3.1 (Term Frequency)**.** Let d be the document, and let i be a term or word in the document, then the *term frequency* of the ith term in the document d is denoted by $\mathrm{tf}(i, d)$ with the following formula: $\mathrm{tf}(i, d) = \frac{|i|}{\sum_{\forall i' \in d} |i'|}$, where i' is any other term other than i.

Definition 5.3.2 (Inverse Document Frequency)**.** Let N be the total number of documents D in the corpus, and let i be the term in d such that $d \in D$, then the *inverse document frequency* is defined by

$$\mathrm{idf}(i, d) = \log \frac{N}{|\{d : d \in D \text{ and } i \in d\}|} \tag{5.4}$$

5.3.2 Word Embeddings

TF-IDF has several limitations, like the initial discussions above, each word will have its own dimension. Therefore, the resulting embedding will be very high in dimension. Further, while TF-IDF can create pairs of words in its bag-of-words to add a little context to the given word, it still cannot capture the full context of the sentence. Lastly, TF-IDF cannot handle out-of-vocabulary words without re-running all the computations to account the new word.

Having said all the limitaitons of the TF-IDF, *word embeddings*, which is understood as based on Neural Network became popular since it addresses all of TF-IDF limitations. First, these word embeddings are dense and lower in dimension compared to the TF-IDF embeddings. Second, these embeddings capture the semantic relationship and similarities between words. Advanced models like Bidirectional Encoder Representation from Transformer (BERT) and Generative Pre-Trained Transformers (GPT) can generate contextualized embeddings, meaning the same word can have different representations depending on its context within a sentence. Both BERT and GPT are discussed in the next chapter. Third, these embeddings have the flexibility to handle out-of-vocabulary words. Lastly, these embeddings pre-trained on large corpora can be fine-tuned for specific tasks.

In summary, while TF-IDF is a straightforward and interpretable method for measuring term importance, it falls short in capturing the semantic richness and contextual nuances of language. Neural network-based word embeddings address these limitations by providing dense, semantically rich, and context-aware representations of words, making them more powerful and versatile for a wide range of natural language processing tasks.

Chapter 6

Methodology

This chapter is organized as follows: Section 6.4.1 will discuss the concept of Topic Modeling, Section 6.4.3 will discuss the concept of Large Language Models, and finally Section 6.7 will discuss how to implement this in Julia programming language.

6.1 Structural Analysis

6.1.1 Bayesian Statistical Models

6.2 Rhythmic Analysis and Modeling

Analyzing the rhyme starts with collection of the ending syllable of the last word of each *aya* أَية in the *sūra* سُورَة. This can be done per *sūra* سُورَة or across the Qur'ān. For this study, both are done to see if there are interesting patterns.

The analysis is done by visualizing the signature of the rhythms, and also modeling the transition probabilities between the ending syllables. That is, answering the question like what is the probability of getting a ending rhyme of *iyn* اِين given the previous verse ends with *uwn* اُون.

Therefore, each *sūra* سُورَة will have its own probabilistic graphical model, which can be analyzed by clustering similar rhymings of *sūra* سُورَة.

The signature can be plotted using line chart, and the verses can be divided into juz. So that, there will be 30 rows for 30 juz.

6.2.1 Bayesian Network Model

6.3 Verse Division Analysis and Modeling

In theory, verse division depends on the rhyme and the length of the verse. For this to be modeled, a discrete-time markov chain can be used with transition probability modeled by a Geometric distribution. The idea is that an *aya* أَية will end if enough number of words have been recited and that the rhyme is achieved. The question is how many number of words to be recited for it to be enough to transition to the next verse. The states here can be defined as the ending rhyming syllable used in the previous section.

6.3.1 Bayesian Discrete-Time Markov Model

6.4 Thematic Analysis

This section

6.4.1 Topic Modeling

As presented in Chapter 1, the first objective is to extract the thematic themes of *sūrahs* سُور with at least 1000 words. In Statistics and Machine Learning, this task is called Topic Modeling. There are several ways to do this, but the popular methodology is to use the Latent Dirichlet Allocation (LDA) discussed in the next section.

6.4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a Statistical methodology that is based on Bayesian inference (Bayes, 1763; Laplace, 1986). It is a generative probabilisitic model for collection of discrete data such as text corpora (Blei, Ng, & Jordan, 2003). The main formula is defined below:

Definition 6.4.1 (Latent Dirichlet Allocation)**.** Let $\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}$ be the random variables, and let α and β be the hyper-parameters, then the probability of generating a document is

$$\mathbb{P}(\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}) = \prod_{j=1}^{m} \mathbb{P}(\boldsymbol{\theta}_j; \alpha) \prod_{i=1}^{k} \mathbb{P}(\boldsymbol{\varphi}; \beta) \prod_{t=1}^{n} \mathbb{P}(\mathbf{Z}_{j,t}|\boldsymbol{\theta}_j)\mathbb{P}(\mathbf{W}_{j,t}|\boldsymbol{\varphi}_{\mathbf{Z}_{j,t}}) \tag{6.1}$$

6.4.3 Large Language Models

Generative Artificial Intelligence or GenAI for short has been making waves on its effectiveness to generate texts, images, audio, video, etc. It has elevated humanity to a new level of capability. However, behind this amazing capabilities is that GenAI is by design a mathematical formula that are called *model*. There are several types of *models*, and one of those is the Large Language Model (LLM). The following section will discuss what LLM is and its mathematical formulation.

6.4.4 Bidirectional Encoder Representation from Transformers

BERT or Bidirectional Encoder Representation from Transformers model is a large language model proposed by Devlin, Chang, Lee, and Toutanova (2018). From the name itself, it is based on the Transformer model architecture (*see* discussion in Section 4.5) in that it only uses the Encoder layer, and stack it together. BERT was pre-trained on large corpus of text using two unsupervised (*see* Section 3.7.4) tasks, and these are:

1. *Masked Language Modeling* (*MLM*) - tokens (*see* Section 5.2) are randomly masked in the input and trains the model to predict these masked tokens based on the surrounding context.

2. *Next Sentence Prediction* (*NSP*) - tains the model to understand the relationship between two sentences by predicting if one sentence follows the other.

After pre-training the model, BERT can then be fine-tuned on specific tasks like question answering, sentiment analysis, and more with relatively smaller datasets. With that, BERT works as follows:

1. *Input Representation* - BERT takes tokenized text as input, which includes a pair of sentences. The input is converted into tokens, added with special tokens like [CLS] (classification token at the beginning) and [SEP] (separator token between sentences).

2. *Embedding Layer* - The tokens are converted into embeddings which are the sum of token embeddings, segment embeddings, and position embeddings.

3. *Encoder Layers* - The embeddings are then passed through multiple layers of bidirectional Transformer encoders (*see* Section 4.5), which apply self-attention mechanisms to generate contextualized representations for each token.

4. *Output* - The final hidden states from the encoder layers are used for different tasks:

    - The [CLS] token's representation can be used for classification tasks.
    - The representations of other tokens can be used for tasks like named entity recognition (NER) or question answering.

There are several applications of BERT model, but for this paper it will be used for Topic Modeling and Text Summarization of the Qur'ān. In particular, CL-AraBERT model by will be used for extracting embeddings of the Qur'ānic words for further analysis.

6.4.5 Generative Pre-Trained Transformer

GPT or Generative Pre-Trained Transformer is another large language model proposed by (Radford, Narasimhan, Salimans, Sutskever, et al., 2018). From the name itself and like BERT, GPT is based on the Transformer model (Vaswani et al., 2017), *see* Section 4.5. Unlike BERT though, GPT uses the decoder layer of the Transformer model and stacks it multiple times. This is the model that is powering the ChatGPT[1] of OpenAI and also Claude AI[2] of Anthropic[3].

GPT models like those powering ChatGPT were pre-trained on large corpora by going through the sequence of the texts in *unidirection*, which is contrary to the *bidirectional* approach of BERT model. As such, the GPT models excel in generating text and performing tasks that require producing coherent sequences of words, in applications like text completion and creative writing. Whereas BERT is technically effective for tasks requiring deep contextual understanding such as text classification and named-entity recognition.

For this paper, the

[1]https://chat.openai.com/
[2]https://claude.ai/
[3]https://anthropic.com/

6.5 Concentrism Formulation

6.5.1 Cosine Similarity

6.5.2 Kullback-Leibler Divergence

6.6 Integrating Other Islamic Literatures

6.6.1 Retrieval-Augmented Generation

The problem with LLM is that it was only trained on huge but limited data, and is therefore not able to infer what should be the context when asked.

6.7 Programming Languages Setup

This section will discuss the coding setup. As mentioned in Chapter 1, the main programming language to use is Julia. As such, it is necessary to present where the codes will be stored so that readers are able to reproduce it. All of the codes will be saved in Github repository accessible through the following link:

`https://github.com/alstat/ma-thesis/tree/main/codes`

6.7.1 Julia

6.7.2 Python

6.7.3 R

Chapter 7

Results and Discussions

This chapter is organized as follows: Section 7.1 will discuss the results of the descriptive statistics; Section 7.2 will discuss the results on the morphological analyses; Section 7.3 will discuss the structural analyses including the Mathematical structures of the Qur'ān; Section 7.4 will discuss the results for thematic analyses using both statistical and Large Language models; and finally, Section 7.5 will discuss the use of other Islamic texts that will help in adding more contexts for the Large Language models.

7.1 Descriptive Statistics

This section will focus on the results of the descriptive statistics of the Qur'ān's *ayāt* ایَات (verses) and *kalimat* کَلِمَات or words. Figure 7.1 visualizes the frequency of the *ayāt* ایَات and words of the Qur'ān using a combinations of bar, density, and box plots. The figure is divided into three main parts. The first part is the statistics of the *ayāt*'s ایَات count. It can be seen that in terms of the number of verses, it is generally decreasing just like what the Muslims observed. Table 7.1 summarizes the necessary statistics of Figure 7.1. From the said table, there are 39 *ayāt* ایَات to expect based on the median statistics, and there are about 55 *ayāt* ایَات to expect per *sūrah* سُورَة based on the mean statistics. The reason the mean is higher than the median follows from the fact that there are surahs that can be considered outlier because of the large number of *ayāt* ایَات. The annotation on the *sūrah* سُورَة with the highest number of *ayāt* ایَات are indicated in Figure 7.1. These *sūrah* سُورَة pushes the mean to higher number than the median. Indeed, these *sūrahs* سُور have stretched the shape of the density and the box plots to higher values, suggesting that the data points are more varied. The width of the density and the box plots is measured by the variance and the standard deviation, which is simply the square root of the variance.

Table 7.1: Descriptive statistics of the *ayāt* ءَايَات counts and the counts of its words

Count Data	Mean	Median	Std. Deviation
Ayahs	54.70	39	53.21
Words	679.20	344	931.18
Words per Ayah	10.27	8.23	6.35

Figure 7.1: Statistics of the words and *ayāt* ءَايَات (verses) of the Qur'ān

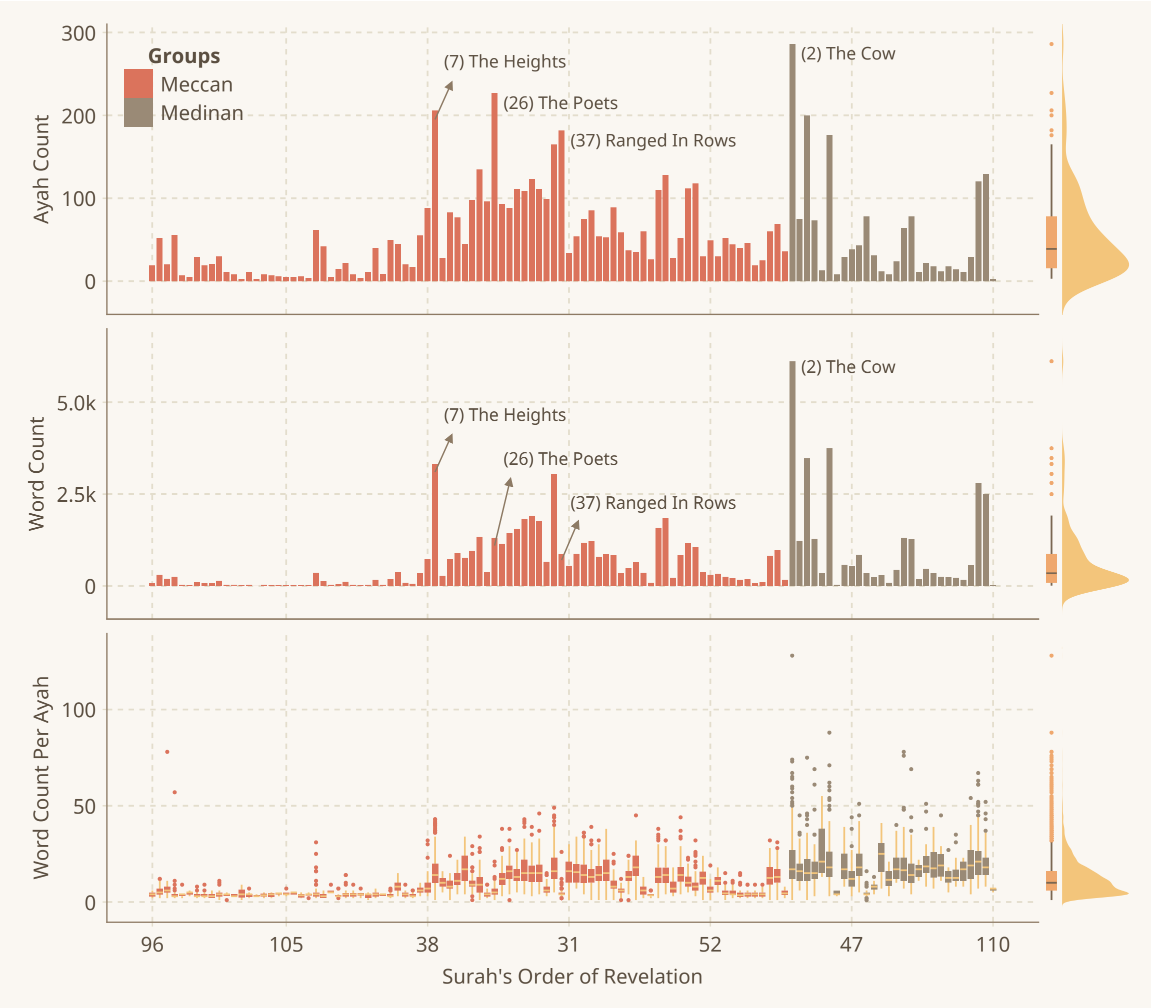


Figure 7.2: Statistics of the words and *ayāt* آيَات (verses) of the Qur'ān according to revelation order

7.1.1 Verses

7.2 Morphological Analysis

7.3 Structural Analysis

7.3.1 Concentric Structure

7.3.2 Mathematical Structure

7.3.3 Discussions on Islamic Philosophy of Qur'ān's Structural Analysis

7.4 Topic Modeling

7.4.1 Latent Dirichlet Allocation

7.4.2 Bidirectional Encoder Representation from Transformer

7.4.3 Generative Pre-Trained Transformer

7.5 Relating to other Islamic Texts and Analyses

7.5.1 Retrieval-Augmented Generation Approach

7.6 Limitations of the Models

Chapter 8

Conclusion and Recommendation

References

Asaad, A.-A. B. (2021). QuranTree.jl: A Julia package for Quranic Arabic corpus. In *Proceedings of the sixth arabic natural language processing workshop* (pp. 208–212). Kyiv, Ukraine (Virtual): Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.wanlp-1.22`

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, *53*, 370-418. Retrieved from `http://www.jstor.org/stable/105741`

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. Retrieved from `https://epubs.siam.org/doi/10.1137/141000671` doi: 10.1137/141000671

Birmingham University. (2015). *Birmingham qur'an manuscript dated among the oldest in the world.* Birmingham University. (Available at: `https://www.birmingham.ac.uk/news-archive/2015/birmingham-quran-manuscript-dated-among-the-oldest-in-the-world` (Accessed: July 8th, 2023))

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Cooperman, A., O'Connell, E., & Stencel, S. (2011). *the future of the global muslim population* (Tech. Rep.). Pew Research Center.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dukes, K., & Habash, N. (2010, May). Morphological annotation of Quranic Arabic. In N. Calzolari et al. (Eds.), *Proceedings of the seventh international conference on language resources and evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). Retrieved from `http://www.lrec-conf.org/proceedings/lrec2010/pdf/276_Paper.pdf`

Karatay, F. E. (1962). *Topkapı sarayı müzesi kütüphanesi arapça yazmalar katalogu.*

kur'an, kur'an ilimleri, tefsirler no. 1 - 2171. Küçükaydın Matbaası, İstanbul.

Khalifa, R. (1981). *The computer speaks: God's message to the world*. Renaissance Production.

Laplace, P. S. (1986, 08). Memoir on the probability of the causes of events. *Statist. Sci.*, *1*(3), 364–378. Retrieved from `http://dx.doi.org/10.1214/ss/1177013621` doi: 10.1214/ss/1177013621

Manila Bulletin. (2022). *PH embassy in riyadh hosts first asian qur'an memorization contest.* Manila Bulletin. (Available at: `https://mb.com.ph/2022/04/30/ph-embassy-in-riyadh-hosts-first-asian-quran-memorization-contest/` (Accessed: July 8th, 2023))

Nassourou, M. (2011). *Using machine learning algorithms for categorizing quranic chaptersby major phases of prophet mohammad's messengership.*

Neuwirth, A. (2007). *Studien zur komposition der mekkanischen suren*. Berlin, Boston: De Gruyter. Retrieved 2023-07-09, from `https://doi.org/10.1515/9783110920383` doi: doi:10.1515/9783110920383

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Sadeghi, B., & Bergmann, U. (2010). The codex of a companion of the prophet and the qurān of the prophet. *Arabica*, *57*(4), 343 - 436. doi: \url{https://doi.org/10.1163/157005810X504518}

Shahzadi, N., ur Rahman, A., & Sawar, M. J. (2012). Semantic network based classifier of holy quran. *International Journal of Computer Applications*, *39*, 43-47. Retrieved from `https://api.semanticscholar.org/CorpusID:6539715`

Sharaf, A., & Atwell, E. (2009). Knowledge representation of the quran through frame semantics: a corpus-based approach. In *Proceedings of the fifth corpus linguistics conference.* The Fifth Corpus Linguistics Conference. Retrieved from `https://api.semanticscholar.org/CorpusID:18278736`

Sharaf, A., & Atwell, E. (2012a, May). QurAna: Corpus of the Quran annotated with pronominal anaphora. In N. Calzolari et al. (Eds.), *Proceedings of the* *eighth international conference on language resources and evaluation (LREC'12)* (pp. 130–137). Istanbul, Turkey: European Language Resources Association (ELRA). Retrieved from `http://www.lrec-conf.org/proceedings/lrec2012/pdf/123_Paper.pdf`

Sharaf, A., & Atwell, E. (2012b, May). QurSim: A corpus for evaluation of relatedness in short texts. In N. Calzolari et al. (Eds.), *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 2295–2302). Istanbul, Turkey: European Language Resources Association (ELRA). Retrieved from `http://www.lrec-conf.org/proceedings/lrec2012/pdf/190_Paper.pdf`

Sidky, H. (2020). On the regionality of qur'anic codices. *Journal of International Quranic Studies Association*. doi: http://dx.doi.org/10.5913/jiqsa.5.2020.a005

Sinai, N. (2014). When did the consonantal skeleton of the quran reach closure? part ii. *Bulletin of the School of Oriental and African Studies*, *77*(3), 509–521. doi: 10.1017/S0041977X14000111

Sinai, N. (2017). *The qur'an: A historical-critical introduction*. Edinburgh University Press Ltd.

Thabet, N. (2004). Stemming the qur'an. In *Proceedings of the workshop on computational approaches to arabic script-based languages* (p. 85-88). USA: Association for Computational Linguistics.

Thabet, N. (2005). Understanding the thematic structure of the qur'an: an exploratory multivariate approach. In *Proceedings of the acl student research workshop* (p. 7-12). USA: Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wansbrough, J. (2004). *Quranic studies: Sources and methods of scriptural interpretation*. Prometheus Books.

Wikipedia. (2015). *Comparison of a 20th-century edition of the quran (left) and the* *birmingham quran manuscript* (*right*). Wikipedia. (Available at: https://en.wikipedia.org/wiki/Birmingham_Quran_manuscript (Accessed: July 9th, 2023))