DISCOVERY OF ANOMALOUS PATTERNS WITHIN MULTIDIMENSIONAL, ASYNCHRONOUS TIME-SERIES WITH AN EMPHASIS ON THE “INTERNET OF THINGS”

by

STEPHEN P. EMMONS JR.

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2014

To my lovely and loving wife Lisa without whose encouragement and support
I could not have completed this work.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Farhad Kamangar who patiently guided me through the unfamiliar waters of the doctoral process, along with my dissertation committee members Dr. Manfred Huber, David Levine, and Dr. Gergely Zaruba for their interest in my research, along with their insight and support.

I would also like to thank Dr. Jeff Smith for his encouragement and moral support, together with the executives at Numerex who provided financial support for my studies and the flexibility to pursue them.

I would thank my colleague Dr. Giacomo Ghidini for his collaboration on several projects integral to the work of this dissertation, and to our collaborators within the graduate program at UTA, including Roberto Coluccio, Gianluca Privitera, and Prathibha Datta Kumar.

Having already dedicated this work to my wife, Lisa, I would also like to thank my children for their cheers and well-wishes throughout this process, and my mother who always lets me know how proud she is, especially now that, at long last, I became a doctor, though perhaps not the kind she was expecting.

Finally, it seems fitting to thank my late father who, through his doctorate, planted the seed that, after so many years, took root and bore fruit in this work. I know he would be proud.

November 4, 2014

ABSTRACT

DISCOVERY OF ANOMALOUS PATTERNS WITHIN MULTIDIMENSIONAL, ASYNCHRONOUS TIME-SERIES WITH AN EMPHASIS ON THE "INTERNET OF THINGS"

Stephen P. Emmons Jr., Ph.D.

The University of Texas at Arlington, 2014

Supervising Professor: Farhad Kamangar

In this dissertation we examine "Internet-scale" systems that present us with multidimensional time-series data characterized by many sources sending symbols at irregular intervals over a common channel. We explore a unique method for the discovery of hidden populations of similar sources and their previously-unknown behavioral patterns, and using these discoveries, reveal anomalous sources and/or time-frames based on their statistical properties. To do so, we employ several well-studied mechanisms, such as k-means and Principle Component Analysis (PCA), and bring to bear analysis tools from other disciplines, such as the use of n-grams and "motifs," that have not previously been considered in these contexts. While applicable to the study of any system whose attributes can map to the generalized model we present, the approach is of particular interest when dealing with large numbers of remote devices that make up the "Internet of Things" (IoT). The method is applied to several discrete layers of cellular wireless communications infrastructure for a diverse set of commercial Machine-to-Machine (M2M) applications where the behavior of the devices was examined as they interacted with carrier network elements. While it is common to study these systems in the context of their application-level duties, the devices' behavior at lower levels of the solution stack has received less attention, and is often poorly understood by either the application engineers or network operators.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Asking The Question

In June 2013, the Internet was estimated to be accessible to over 2.4 billion of the world's 7 billion humans [1]. But the Internet is never directly used by humans. Instead, humans use some type of computing device – desktops, laptops, smartphones, tablets – to access the Internet. And when connected to the Internet, these devices communicate directly or indirectly with many other types of devices that assist them in the performance of their tasks – servers, routers, and firewalls – or with which they interact – home automation/security devices, video cameras, medical devices, and vehicles.

When some human is not using these computing devices, they are often still connected to the Internet. If not powered off, they can, and very often do, continue performing functions of which the human user may only be vaguely aware. As an everyday example, the casual observer of a common consumer WiFi router will notice that its many status lights are almost always flickering to indicate a constant level of activity by nearby connected devices, and might ask the question: "What are those *things* doing?"

One might say that the Internet is really made up of "things" and that humans are simply hovering around the edges. In fact, there are many more "things" that use the Internet than humans. Kevin Ashton, co-founder of the Auto-ID Center at MIT, is credited with first using the phrase "The Internet of Things" (IoT) in 2009. Cisco's Dave Evans estimates that the IoT was "born" around 2008 when the number of “things” connected to the Internet exceeded the number of humans, and further estimates that they will exceed 25 billion by 2015 and then 50 billion by 2020 [2].

With so many “things,” a better question might be: “What are *all those things* doing??”

The challenge for my research is to identify very specific ways to discover what *all those things* are doing by examining the flow of information they produce at key observation points within the Internet, cross-referencing that with other information known about them, and bringing to light previously unknown or unclear information about their condition and/or behavior.

1.2 Framing The Problem

The large and growing size of the IoT results from the opportunity provided by the Internet to collect data about almost anything, anywhere, anytime, and then send that data almost anywhere, anytime.

The commoditization of microprocessors has made it possible to create both dense and widely-dispersed sensor networks used in the monitoring and control of everything from security and surveillance systems, HVAC systems, appliances, all types of vehicles, electric and other utility distribution/metering, medical devices, and growing array of biometric devices to enhance the human experience.

With the ability to embed microprocessors connected to sensors in so many places to collect data comes the need to deliver it somewhere that it can be used most effectively. The Internet, with its global wired infrastructure augmented with cellular, satellite, and other wireless networks, has become the medium of choice for the delivery of this data. And the destinations are often aggregation points where the data is processed, analyzed, and archived for many different purposes.

Specific populations of embedded devices serving a given purpose, often with one or more aggregation points for monitoring and control, are commonly called "Machine-to-Machine" or M2M applications (Section 2.1). Such applications can range in size from the hundreds to the millions of devices. In many cases, very few humans are involved in the day-to-day operations of such systems. Thus, the IoT is able to grow unconstrained by the number of humans who also use the Internet.

The designers, developers, and operators of M2M applications may instrument and examine individual devices up to the point that they send data over the Internet, and again at some data egress point from the Internet, such as a data center; however, everything that happens in between is virtually unknown to them. And the methods and access available to look deeply into the information flow to better understand its behavior are limited. These applications' device communications are aggregated together with that of countless others by their service providers.

On the other hand, Internet providers for M2M applications have their own monitoring of the data flow for all of their customers and generally know what traffic belongs to each. Yet they typically have no specific knowledge of the significance or importance of their available information apart from obvious conditions related to outages and congestion. And while Internet providers have expanded their monitoring of human-related traffic in many application-related areas such as spam filtering, intrusion detection, and denial-of-service (DoS) attack mitigation, they currently have little monitoring specific to IoT-related concerns.

Thus, both M2M applications and Internet providers can benefit from a better understanding of the behavior of the many "things" that they must support.

1.3 Laying The Foundation

First, it is important to have a more thorough understanding of the characteristics of the environment within which both M2M applications and Internet providers operate.

In Chapter 2, we start by describing the M2M application domain in greater detail (Section 2.1), reviewing related work in this area, followed by an exploration of key foundational concepts used throughout the remainder of this dissertation in Section 2.2.1 and following.

We consider a form of dimensionality reduction through the creation of "meta-dimensions" based on the identification of temporal sequences of data generated by individual devices using a combination of "n-gram" (Section 2.3.1) and "motif" (Section 2.3.2) techniques.

We survey several methods of clustering that may be used to discover associations within data, with a special focus on k-means, focusing on how to ensure that the data is properly conditioned to yield meaningful results and leveraging the survey of data properties performed earlier (Section 2.3.3).

We further consider ways to reduce the dimensionality of raw available data, focusing specifically on several methods for filtering, normalizing, and then "whitening" the original data using Principal Component Analysis (Section 2.3.5).

With this foundation, we are able to focus on the unique contributions of this dissertation.

1.4 Attacking The Problem

The goal of this research is to look deeply into the flow of data available in the IoT and discover previously unknown or unclear information about the behavior of its denizens. I seek to reach this goal by accomplishing the following objectives:

1. Propose a set of methods to analyze time-series data sources such as those available in the IoT that characterizes their dimensionality, performs various methods of dimensionality reduction and enhancement, and uses clustering methods to identify groupings of both devices and time periods (Chapter 3).
2. Apply this approach to several real-world data sources and assess which of the various forms of the method produce the most useful results (Chapter 4).

The proposed methods go beyond the prior work by considering unique combinations of foundational elements for pattern discovery. In particular, they introduce novel variations of both n-grams and motifs helpful for the real-world data sets being studied.

Several of the real-world scenarios included here have yet to be published outside of this dissertation (Sections 4.2 & 4.3). Included in these studies is the use of "Big Data" collection and analysis methods that were material in conducting much of the experimentation [67] and [68].

1.5 Assessing The Results

At the conclusion (Chapter 5), I intend to show that these methods may be used to discover previously-unknown patterns specific to the systems being studied. My expectation is that this information will then be useful for historical trend analysis of the past, anomaly detection in the present, and/or forecasting expected system state into the future; however, these resulting applications are not my focus. These patterns should reveal important operational dynamics of the many devices interacting within the IoT that will aid the designers, developers, administrators, and operators of M2M and other applications to better understand what *all those things* are doing, and with that understanding, improve existing systems, as well as more successfully plan, build, and operate future systems.

CHAPTER 2

BACKGROUND

2.1 "Machine-to-Machine" Applications

M2M applications bring together a wide array of technologies. They likewise demand an equally-wide range of expertise to properly design, build, test, deploy, and operate them. End-to-end implementation for such a system may involve any or all of the following:

- Specialized sensors requiring calibration.
- Embedded microprocessors with customized firmware that may need future remote updates.
- Battery life or other power management considerations.
- Wireless and other communications needs, with special attention to scalability, routing, and even cost considerations.
- Centralized servers for collecting, processing, analyzing, and archiving sensor data with scalability, reliability, and disaster recovery considerations.
- Human user interface needs for system administration and end-user access to the information for reporting, notification, and other purposes.

Once completed, an M2M application should be thoroughly monitored to ensure proper ongoing operation. Many aspects of the overall system may be monitored using standard measures such as CPU, memory, storage, and communications levels available for centralized servers and networking equipment. Software components may be specifically instrumented during development to make other key operational measures available through log files or by providing standardized access methods such as SNMP or JMX. Many established commercial and open source monitoring systems exist that can capture this information and check for anomalies in order to alert system operators to investigate.

Unfortunately and despite best efforts, this approach to monitoring an M2M application is never enough. A monitoring system can only detect what it was designed to find. Almost certainly, a complex M2M application that grows in size over time will start to exhibit behavior that was not anticipated by the original designers and whose causes are unclear. Invariably, there are clues within the observable data about the overall application that may be found. These may be used to make some corrective or adaptive change to one or more components, and likewise may be used to enhance the monitoring system to detect future similar circumstances.

The longer after the original deployment of the M2M application that such issues arise, the more challenging they become. Due to the diversity of expertise required to build and run an M2M application, it is difficult in practice for any one person to fully grasp the complexity involved in the whole system. Some on the original design and development team may have such a grasp, but they may not be involved with the day-to-day operations after a field trial. All knowledge by system administrators and operators will likely derive from initial training plus their own observation of limited parts of the system to which they have access. As a result, the people who are the closest to the system – the administrators and operators – are often the least qualified to understand the significance of what they see happening. But eventually such issues should be escalated to the the original development team, assuming they are available, to take the necessary action.

Perhaps the most challenging aspect of an M2M application is the very use of the Internet as the communications medium for its many remote devices. As the population of devices grows, so do the number of variables that can influence their behavior. Over time the devices may consist of units from different manufacturing runs, may have slightly different firmware or configuration settings, may have different remaining battery life due to both age and use, may be in different geographic locations, or may be suffering from some network-related issue within the Internet itself; all such factors may affect device behavior and be reflected in the aggregate data flow to and from them.

Before M2M applications, there were SCADA systems and other telemetry systems. Current methods reflect the successful approaches used in earlier or similar systems that are brought to bear on M2M systems with limited success. Traditional systems SCADA and telemetry systems are closed and collect data at regular, high sampling rates. Methods for analyzing such data borrows heavily from signal processing. Monitoring, analysis, and visualization tools from these systems are mature and well-understood.

In contrast, M2M applications tend to generate very low-frequency, even "undersampled," data by comparison to minimize often-costly communications requirements. Concerns for possible spikes in capacity demand lead many system designers to design and Internet providers to require the randomization of communications from large populations of devices, the resulting aggregate data stream often requires conditioning to prepare it for use in traditional tools.

M2M applications, and the specific aspects of the Internet infrastructure on which they rely, have become growing subjects of research. Areas of particular focus are performing case studies on real-world systems in general [3] and in specific (e.g., vehicle tracking [4, 5] and energy "smart grids" [6, 7]), developing standards to benefit future solutions [8, 9, 10], measuring quality of service (QoS) for greater accountability between M2M applications and Internet providers [11, 12, 13], simulating scale to anticipate future scenarios [14, 15], planning for and investigating future infrastructure needs [16, 17, 18], and optimizing various aspects of systems operations [19, 20, 21, 22].

In the literature, researchers have started to distinguish M2M from "Human-to-Human" or H2H system characteristics. Some H2H systems, such as "Twitter," share many common characteristics with M2M systems and may be considered part of the IoT. Relevant research in this area includes identifying trends [23] and classifying human actors [24].

2.2 Time-Series Basics

Before proceeding further to consider various foundational concepts and mechanisms used in the analysis of time-series data from systems like M2M applications, we must first establish a common notation and constructs to use throughout the rest of this dissertation.

2.2.1 Notation

Before proceeding further to consider various foundational concepts and mechanisms used in the analysis of time-series data from systems like M2M applications, we must first establish a common notation to use throughout the rest of this dissertation.

Suppose we have a set of information sources S which have sent a set of messages M to a set of destinations D over some period of time. Each message μ in M may be composed of many elements and the variety of messages in M fall within a larger message domain $\mathbb{M}$ such that $M \subseteq \mathbb{M}$.

Within this larger message domain $\mathbb{M}$, there is a set of functions $\mathbb{V}$ that can be used to extract information values from M. Many message domains have functions returning the time t when the message occurred ($\mathcal{V}_T(\mu) = t$), plus the source and destination of the message ($\mathcal{V}_S(\mu) = s$ where $s \in S$ and $\mathcal{V}_D(\mu) = d$ where $d \in D$).

Other functions may exist to extract discrete or continuous values from M. For example, one function $\mathcal{V}_{sym}(\mu) = v_{sym}$ may produce a value v_{sym} that must be one of n discrete symbols $sym_1 \dots sym_n$. Another function $\mathcal{V}_{0:1}(w) = v_{0:1}$ may produce any real number $\mathbb{R}$ between 0 and 1 inclusive.

Within this framework and depending on the domain, additional external constraints or conditions may exist that limit the possible content found in a message μ of M. For example, the values of t produced by $\mathcal{V}_T(\mu)$ may occur at some fixed interval ($t_{i+1} = t_i + \Delta t$), may allow multiple messages to occur at the same time ($t_{i+1} \geq t_i$), or may disallow it ($t_{i+1} > t_i$). Furthermore, the possible sources S may be the same as the possible destinations D ($S = D$), may be mutually exclusive ($S \cap D = \{\}$), or may have very different cardinality ($||S|| \gg ||D||$ or $||S|| \ll ||D||$).

When working with time series where the information values are discrete and countable, it is often desirable to count the frequency of occurrence of certain values found within M or subsets of M. For notational convenience, we will use subscripts and superscripts to characterize different subset of M by some criteria. Thus we may say either $M_x = \{\mu \mid \mathcal{V}_X(\mu) = x\}$ or $M^{(y)} = \{\mu \mid \mathcal{V}_Y(\mu) = y\}$, and use $||M_x||$ or $||M^{(y)}||$ as the count of the number messages μ in M_x or $M^{(y)}$, respectively. We may also say $M_x^{(y)} = \{\mu \mid \mathcal{V}_X(\mu) = x \wedge \mathcal{V}_Y(\mu) = y\}$, with $||M_x^{(y)}||$ as the count of messages μ having both criteria. Note that due to the use of superscripts, if a transpose or inverse matrix is intended, we will denote these with parentheses (e.g., $(M_x^{(y)})^{\mathrm{T}}$ or $(M_x^{(y)})^{-1}$, respectively).

Since M_x is a set and given that x is some discrete value of another set X such that $x \in X$, it is convenient to consider M_X as a vector of sets and $||M_X||$ as a vector of counts. For this to be possible, we must implicitly assume an ordering of the elements of X that are not, strictly speaking, defined. However, if $||X|| = n$ and

$n \neq 0$, then we can declare that the vector $v = ||M_X||$ has size $||v|| = n$, and the values of v_i (or $v(i)$) where $i = [1..n]$ correspond to the values of $||M_{x(i)}||$ for all i.

Furthermore, we may consider $||M_X||$ to be a *horizontal* vector, and similarly, $||M^{(Y)}||$ to be a *vertical* vector of size $m = ||M^{(Y)}|| = ||Y||$. And finally, it follows that $V = ||M_X^{(Y)}||$ may be a rectangular matrix of size $n \times m$ where $V(i,j) = ||M_{x(i)}^{(y(j))}||$.

$$V = \begin{bmatrix} ||M_{x(1)}^{(y(1))}|| & \cdots & ||M_{x(i)}^{(y(1))}|| & \cdots & ||M_{x(n)}^{(y(1))}|| \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ ||M_{x(1)}^{(y(j))}|| & \cdots & ||M_{x(i)}^{(y(j))}|| & \cdots & ||M_{x(n)}^{(y(j))}|| \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ ||M_{x(1)}^{(y(m))}|| & \cdots & ||M_{x(i)}^{(y(m))}|| & \cdots & ||M_{x(n)}^{(y(m))}|| \end{bmatrix} \tag{2.1}$$

2.2.2 Aggregate Time Windows

When working with time series, it is common to aggregate the messages of M into subsets based on a sequence of time intervals, or “windows,” W. As mentioned earlier, actual time values t extracted from M may or may not be subject to external constraints, particularly for “real world” data that may occur at any time. It is important, therefore, to precisely define the time intervals used to partition the larger message set M.

A sequence of time intervals W necessarily has a size $m = ||W||$ with the intervals defined as $[t_{j-1}, t_j)$ for $j = [1 \dots m]$, and where the first interval is $[t_0, t_1)$ and the last is $[t_{m-1}, t_m)$ and the entire sequence of intervals is bounded by $[t_0, t_m)$. Typically, there exists a fixed spacing between the values of t such that $t_j - t_{j-1} = \Delta t$ holds for all intervals. In the notation, we may refer to time intervals either by their starting times $t_0 \dots t_{m-1}$ or by their sequence order $w_1 \dots w_m$.

The values of m, t_0, t_m, and Δt may be chosen in several ways depending on the circumstances. A given message set M naturally has t_{min} and t_{max} values corresponding to the smallest and largest values for t found for any message μ in M.

One obvious choice would be to declare m as some specific number (e.g, 10), set $t_0 = t_{min}$, and $\Delta t = (t_{max} - t_{min})/m$ to fully utilize the entire message set M.

However, a more common choice is to choose some t_0 representing the start of an important overall time period to be analyzed (e.g., the start of a specific day), and to likewise choose a Δt to be period matching some practical duration of time for analysis (e.g., one minute or one hour).

Note that such choices may either exclude messages in M where $t_0 > t_{min}$ or $t_m < t_{max}$, or result in partitioned sets of $M^{(W)}$ where $M^{(w_1)}$ and/or $M^{(w_m)}$ are possibly under-represented as compared to other sets $M^{(w_j))}$ when considering attributes such as the counts of $||M^{(W)}||$. These are considerations for an analyst working with W.

2.2.3 Probability

As we shall see later, probability distributions play an important role in the detection of patterns with time series. Thus it is important to identify some important distributions and briefly discuss their properties.

From a frequentist perspective and considering the sources S, we may calculate the probability of messages in M being sent from any source s in S, or $P(S)$, as follows:

$$P(S) = \{p_s = \frac{||M_s||}{||M||} \mid s \in S\} \tag{2.2}$$

By extension, we may define the distribution for any of the functions $\mathbb{V}$ and set of messages M. Using the previous example function $\mathcal{V}_{sym}$, the distribution P_{sym} is simply. . .

$$P(sym) = \{p_{sym} = \frac{||M_{sym}||}{||M||} \mid sym \in sym_1 \dots sym_n\} \tag{2.3}$$

Next, a somewhat different distribution may be obtained for an arbitrary subset of M for time interval w_0 called $M^{(w_0)}$.

$$P^{(w_0)}(S) = \{p_s^{(w_0)} = \frac{||M_s^{(w_0)}||}{||M^{(w_0)}||} \mid s \in S\} \tag{2.4}$$

Again for notational convenience, we may consider discrete distributions like $P(S)$ and $P^{(w_0)}(S)$ as also representing *horizontal* vectors of size n where $||S|| = n$.

$$P(S) = [p_{s(1)} \cdots p_{s(i)} \cdots p_{s(n)}] = \frac{||M_S||}{||M||} \tag{2.5}$$

$$P^{(w_0)}(S) = [p_{s(1)}^{(w_0)} \cdots p_{s(i)}^{(w_0)} \cdots p_{s(n)}^{(w_0)}] = \frac{||M_S^{(w_0)}||}{||M^{(w_0)}||} \tag{2.6}$$

Given that $P^{(w_0)}(S)$ defines a *horizontal* vector, it follows that $P^{(W)}(S)$ defines a matrix with each row being a corresponds the $P^{(w(j))}(S)$. Note that this matrix is different than the *joint* probability distribution for S and W which might also be represented as a matrix, but where the individual elements would represent the probability of the occurrence of each unique combination of s_i and w_j. As we shall see, maintaining separate distributions as the rows in a matrix will be helpful later when analyzing time series.

2.2.4 Entropy And Divergence Measures

When working with many probability distributions such as those defined by $P^{(W)}(S)$, it can be helpful to compare them to one another in different ways.

On way to compare distributions is to compute their "Shannon Entropy," or H, and use this as a metric [25]. The standard calculation for H_S given some distribution $P(S)$ is as follows

$$H_S = -\sum_{s \in S} p_s \log_2(p_s) = -\sum P(S) \log_2(P(S)^{\mathrm{T}}) \tag{2.7}$$

Given $P^{(W)}(S)$, we can produce the a *vertical* vector $H^{(W)}(S)$.

$$H^{(W)}(S) = -\sum_{S} P^{(W)}(S) \log_2(P^{(W)}(S)^{\mathrm{T}}) \tag{2.8}$$

Entropy represents a measure of the "uncertainty" present in the probability distribution. Values for H_S may range from 0 to $\log_2(||S||)$.

A value of 0 occurs when there is *no* uncertainty in the distribution, or said another way, when only one possibility exists, having a probability of one (1) and all others being zero (0). In such cases, the $1 \times \log_2(1) = 0$ and $0 \times \log_2(0) = 0$. Thus the sum of all such terms is likewise zero (0).

The maximum possible value of $\log_2(||S||)$ occurs the occurrence of any symbol of S is equally likely, making the probability $1/||S||$ and each term $(1/||S||)\log_2(1/||S||)$ added together $||S||$ times.

Comparing values of H for different distributions can distinguish between those having differing levels of uncertainty, but similar values of H does not imply that the distributions are similar.

For a more direct comparison of distributions, we may may measure how widely they diverge from one another using Kullback-Leibler Divergence (KLD) [26]. Given two distributions $P = \{p_1 \dots p_n\}$ and $Q = \{q_1 \dots q_n\}$, the definition of KLD is as follows.

$$KLD(P, Q) = \sum_{i=1}^{n} p_i \log_2 \left(\frac{p_i}{q_i}\right) \tag{2.9}$$

It is worth that, by convention, any terms having $q_i = 0$ are ignored, since the $\log_2$ term would therefore be undefined. One way to avoid this problem during calculations is to add a minimal ϵ to each probability p or q to avoid the division by zero with the expectation that the result remains the same within some acceptable precision.

This is also sometimes called the Kullback-Leibler *Distance*, but strictly speaking, it is not a true distance measure. The most critical reason for this is that it is not a symmetric calculation. In other words, it is possible that $KLD(P, Q) \neq KLD(Q, P)$. However, it is true that $KLD(P, P) = 0$ since each term in the sum is multiplied by $\log_2(1) = 0$, and in general, KLD has been found to be a good metric for ranking tasks.

In our later work, we will focus on ensuring that divergences are calculated for specific distributions P against some mean for overall family of distributions Q. In this way, we guarantee that any divisor $q_i > 0$.

The lack of symmetry was initially noted by Kullback and Leibler themselves, leading them to combine the two permutations in their work – $KLD(P, Q) + KLD(Q, P)$ – to eliminate the sensitivity to the order of P and Q.

An alternative to KLD called the JensenShannon Divergence (JSD) solves the symmetry problem by finding the divergence for distributions P and Q from their mean $(P + Q)/2$ and averaging the sum as follows [27].

$$JSD(P,Q) = \frac{1}{2}KLD\left(P, \frac{(P+Q)}{2}\right) + \frac{1}{2}KLD\left(Q, \frac{(P+Q)}{2}\right) \tag{2.10}$$

Note that this equation similarly ensures that the divisor in the $\log_2$ term of KLD is never zero (0) unless $p_i = q_i = 0$. As compared to the simple summing of the permutations $KLD(P,Q) + KLD(Q,P)$, JSD has the advantage of producing values in the same scale as KLD such that they may be compared as necessary. We will use this as needed to compare individual distributions in the later parts of this dissertation.

2.3 Building Blocks

Having established a notation for representing various aspects of any time series in vector and matrix form, we may begin applying various statistical mechanisms for analyzing them.

2.3.1 *n*-Grams

One important mechanism when working with time series with functions returning a discrete value, such as symbol from a set A where $\mathcal{V}_A(\mu) \in A$, is the n-gram.

The use of n-grams is found in many disciplines: They first appeared in Shannon's seminal work the theory of communication [25]. While n-grams were first studied using the Latin alphabet with many studies performed using English [28], they easily apply to both written and spoken language by simply deciding on the symbol set to use. Many examples can be found for Arabic text classification [29], Japanese speech recognition [30], and textual classification of Greek and Chinese [31].

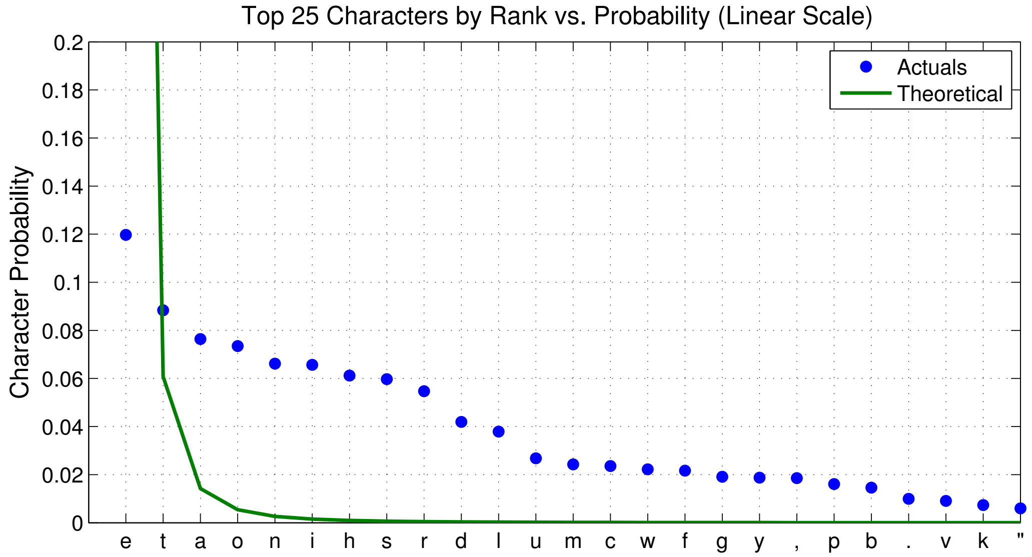


Figure 2.1. The chart shows the top 25 character rankings by probability in X; the Y axis shows the actual probability for each of the top characters, along with a reference line indicating a theoretical hyperbolic distribution; a linear scale is used to highlight the dramatic drop-off in probabilities consistent with a hyperbolic distribution.

In genetics they are used to help classify gene sequences in long strands of DNA where complex patterns of amino acids – commonly labelled A, C, G, and T – may be analyzed using many different methods [32, 33].

They are commonly in featured web-based analytics, most commonly using word sequences where differences in word-sequence frequencies assist with classification techniques [34, 35, 36].

Various schemes for identifying music through the discretization of both monophonic and polyphonic audio tracks into symbol sequences have been based on n-grams [37, 38]

By assigning symbolic labels to frequent geographic locations, n-grams have been used to identify patterns of movement by users of smartphones with GPS capabilities [39].

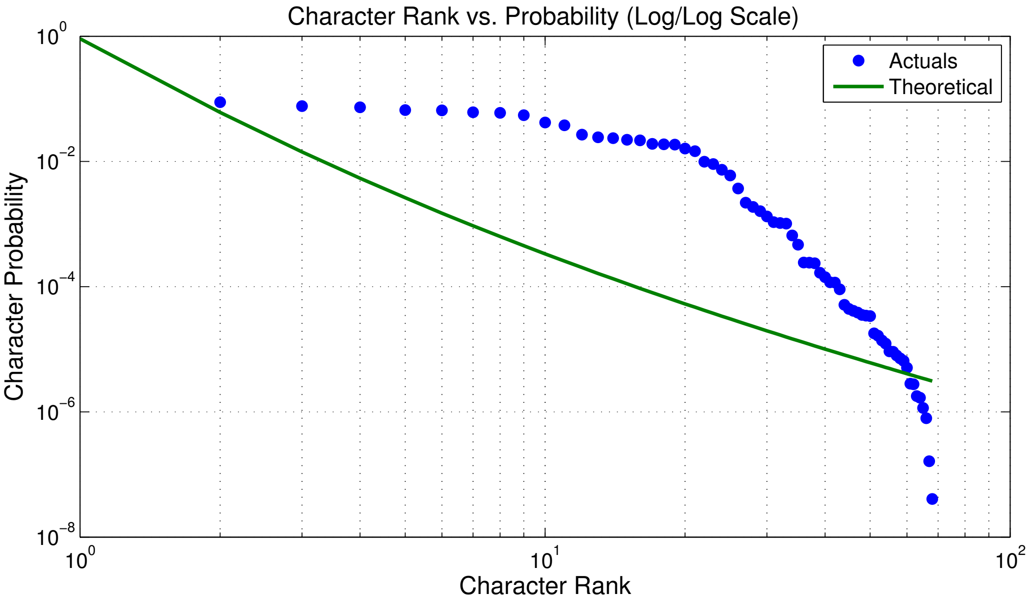


Figure 2.2. The chart shows all characters in descending rank order by probability in X; the Y axis shows the actual probability for each character, along with a reference line indicating a theoretical hyperbolic distribution; the log/log scale us used to illustrate the property of hyperbolic distributions to appear linear on such scales.

An n-gram is an ordered sequence of symbols of length n from some set, or *alphabet*, A. Typically, n-grams identify occurrences of such symbol sequences within some large *corpus* of data. The domain of possible n-grams, or $\mathbb{G}_n$, is simply the permutations of length n of the symbols in A, and as such, grows exponentially with a maximum size $||\mathbb{G}_n|| = ||A||^n$. However, it is often true that the actual number distinct n-gram occurrences, or G_n, found in a given *corpus* is far smaller such that $||G_n|| \ll ||\mathbb{G}_n||$.

The theory behind n-grams have been widely studied. Values for n can be any positive integer starting with the 1-gram, which is simply the starting alphabet A. But given the exponential domain size, practical applications rarely involve a values of n beyond 4 or 5 [38, 31].

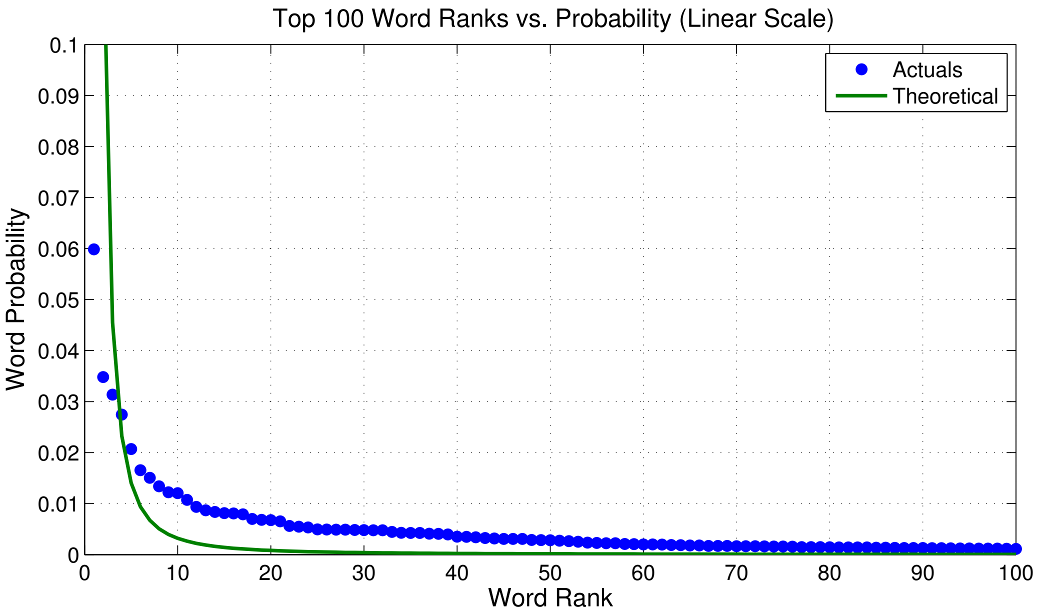


Figure 2.3. The chart shows the top 100 word rankings by probability in X; the Y axis shows the actual probability for each of the top words, along with a reference line indicating a theoretical hyperbolic distribution; a linear scale is used to highlight the dramatic drop-off in probabilities consistent with a hyperbolic distribution.

As mentioned above, the most common applications for n-grams have involved various kinds of textual analysis, with the symbol sets being either characters or words. Often, and indeed in the case studies found later, the n-gram frequencies for these systems follows "Zipf's Law," and as such, fall within the family of *hyperbolic* distributions [40, 32].

To illustrate the characteristics of n-grams, consider the example data set described in Table 2.1. The documents in this data set contain public domain works in English, German, Spanish, and Portuguese and representing a wide variety of genres, including drama, fiction, philosophy, poetry, and children's stories.

Focusing on the English works, it is easy to visualize the hyperbolic frequency distribution found in the data, whether considering characters or words. Figures 2.1

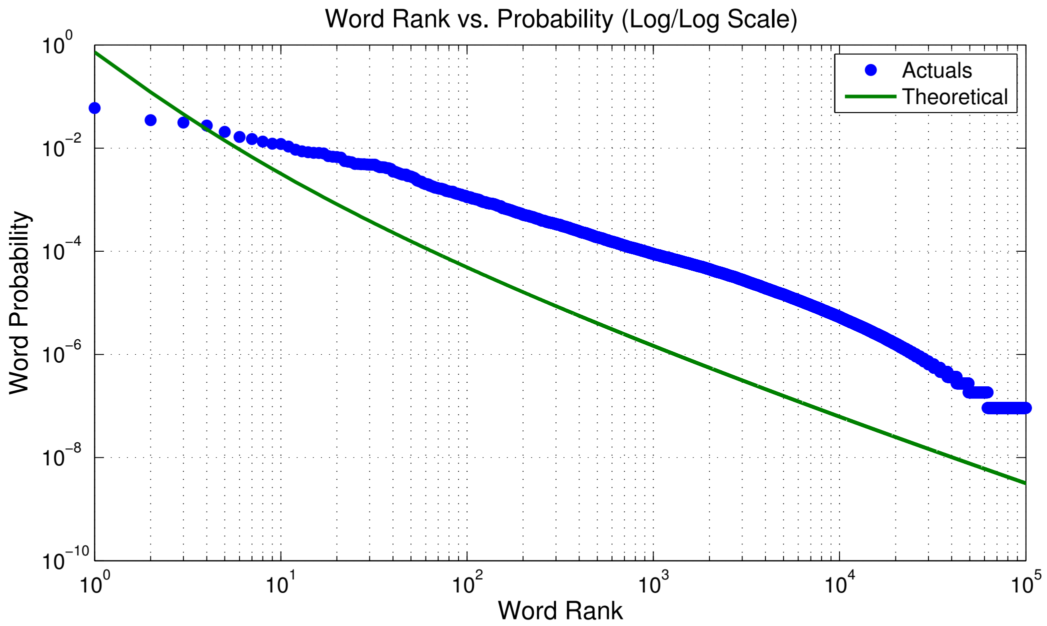


Figure 2.4. The chart shows all words in descending rank order by probability in X; the Y axis shows the actual probability for each word, along with a reference line indicating a theoretical hyperbolic distribution; the log/log scale us used to illustrate the property of hyperbolic distributions to appear linear on such scales.

& 2.2 for character occurrences and Figures 2.3 & 2.4 for word occurrences highlight the frequency extrema seen in such data sets.

Figure 2.1 shows the actual letters in the top 25 rankings, confirming the common empirical observation that the letters ‘e’ followed by ‘t’ are the most common letters in English. The full set of characters shown in Figure 2.2 falls short of matching a hyperbolic distribution due to its relatively small set size.

Figure 2.3 does *not* show the top-ranked words due to space considerations on the chart, but the first several are ‘the’, ‘and’, ‘of’, ‘to’, and ‘a’ which likewise are top rankings expected from English text. The full set of words found in the data are shown in Figure 2.4; however, unlike Figure 2.2, the larger number of words provides

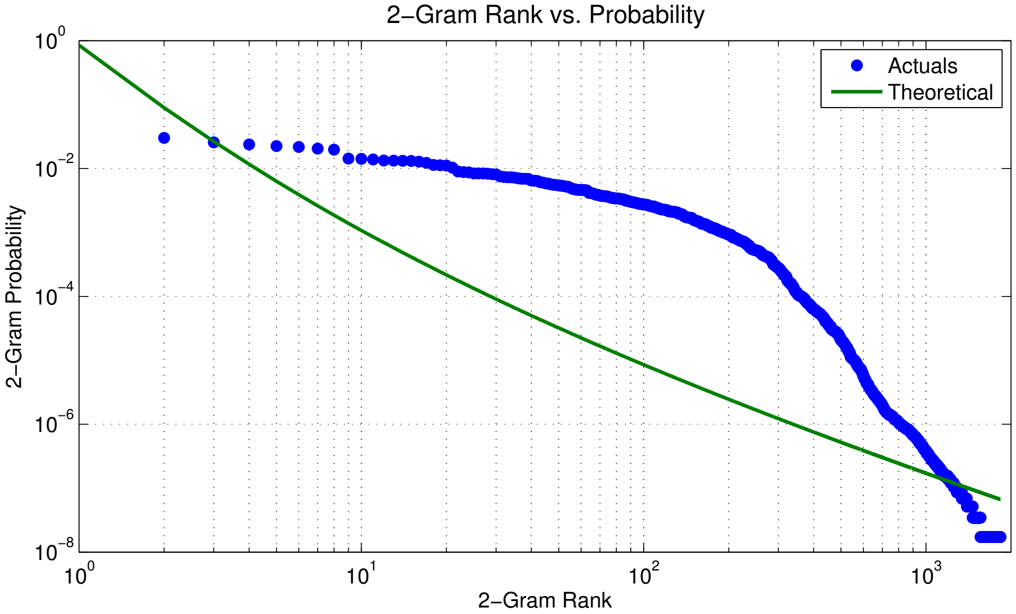


Figure 2.5. The chart shows all 2-grams in descending rank order by probability in X; the Y axis shows the actual probability for each 2-gram, along with a reference line indicating a theoretical hyperbolic distribution; the log/log scale us used to illustrate the property of hyperbolic distributions to appear linear on such scales.

stronger evidence of the linear appearance of hyperbolic distributions plotted on the log/log scale.

It is considered an empirical fact that language symbol frequencies follow such hyperbolic distributions, but it is interesting to note that this behavior is observed most clearly at the limit [41]. Given the small number of "rankable" characters, together with the relatively small data set size of our example, the actual occurrences of characters on the log/log scale only approximates the expected linear placement on the chart. On the other hand, the chart of "actuals" for the much larger collection of words more clearly appear linear on the log/log scale.

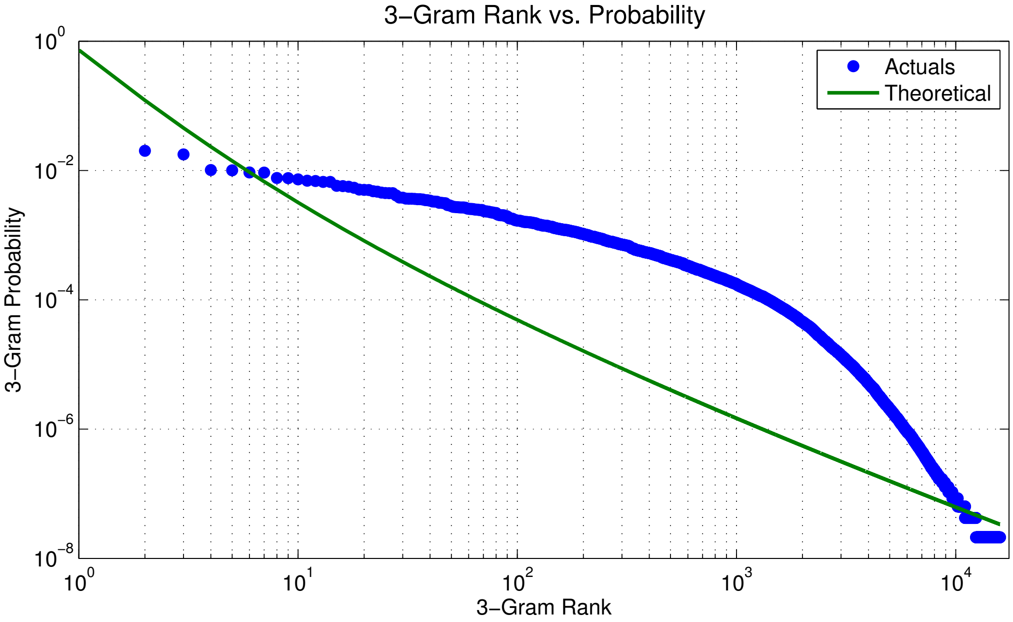


Figure 2.6. The chart shows all 3-grams in descending rank order by probability in X; the Y axis shows the actual probability for each 3-gram, along with a reference line indicating a theoretical hyperbolic distribution; the log/log scale us used to illustrate the property of hyperbolic distributions to appear linear on such scales.

In each of the charts of Figure 2.1 through 2.4, the “theoretical” line is proportional to the frequency $F(r)$ where r is a ranking from 2 to $||A|| + 1$ and c is a constant such that $c > 3$:

$$F(r) = r^{-1} \log(r)^{-c} \tag{2.11}$$

An important difference between character vs. word frequencies is that, while finite, the domain of words used in any living human language is constantly growing, and their frequencies of use changing as well, whereas the character symbol set is constant and comparatively small. From an example standpoint, the handling of characters is more closely related to what one might find when dealing with discrete time series values.

Table 2.1. Example Document Data Set

Attribute	Count
Documents	140
Lines	1,306,008
Words	10,905,526
Characters	59,597,048
Unique Words	109,984
Unique 1-grams	68
Unique 2-grams	1,837
Unique 3-grams	15,922

Table 2.2. Top 10 Example n-grams By Rank

Rank	$n = 1$	$n = 2$	$n = 3$
1	‘e’	‘e ’	‘ th’
2	‘t’	‘ t’	‘the’
3	‘a’	‘th’	‘he ’
4	‘o’	‘he’	‘nd ’
5	‘n’	‘d ’	‘ an’
6	‘i’	‘ a’	‘and’
7	‘h’	‘s ’	‘ed ’
8	‘s’	‘t ’	‘ to’
9	‘r’	‘in’	‘ of’
10	‘d’	‘n ’	‘of ’

Continuing with characters, we observe that the characters themselves are the “1-grams” in our example, of which our data set contains 68 distinct, *case-insensitive* characters. This number could grow if we distinguished between upper- and lowercase letters, but we chose case insensitivity for simplicity.

A “2-gram” is simply an adjacent pair of characters from the set of 68 possibilities. Table 2.1 shows that we found 1,837 unique character pairs out of over 59 million total occurrences. Note that the occurrences of 1-grams, 2-grams, or n-grams for any value of n are essentially the same; for example, the text sequence ‘ the ’ contains

Table 2.3. Example n-gram Entropic Dimensions

n	$\|\|\mathbb{G}_n\|\|$	$\log_2(\|\|\mathbb{G}_n\|\|)$	$\|\|G_n\|\|$	$\log_2(\|\|G_n\|\|)$	$H(P(G_n))$	H/H_{max}
1	68	6.09	68	6.09	4.38	72%
2	4,624	12.18	1,837	10.84	7.45	61%
3	314,432	18.26	15,922	13.96	9.87	54%

five(5) 1-grams – ‘ ’, ‘t’, ‘h’, ‘e’, and ‘ ’ – and four(4) 2-grams – ‘ t’, ‘th’, ‘he’, ‘e ’. This difference of one(1) occurs at the boundary of the total original sequence. As the sequence length grows, the difference in the total 1-grams vs. 2-grams continues to differ only by one(1). Thus the difference in the number of 1-grams vs. n-grams in any data set of containing s sequences is exactly $s \times (n-1)$.

Similarly, a “3-gram” is an adjacent trio of characters from the same set of 68 possibilities of which 15,922 unique permutations were found. The “2-grams” with 1,837 unique pairs are only about 40% of the 4,624 permutations possible, and the “3-grams” seen are only 5% of the 314,432 possible permutations.

It thus follows that collecting counts for messages of a time series where $m = ||M||$ and $a = ||A||$ of all n-grams from $1 \dots n$ can be performed with a time complexity of $O(m \times n)$, and using hash-based indexing of the various n-grams occurrences, with space complexity of $O(a \log(a)^{(n-1)})$.

Figure 2.2 shown earlier, together with Figures 2.5 & 2.6, illustrate that the frequencies of various levels of n-grams, from 1 through 3 respectively, all adhere to the hyperbolic distribution model. And while there is no clear proof that this kind of model will be true for all n, it has at least been empirically observed consistently in research to date [32].

For practical reasons, only Figure 2.1 shows the actual 1-grams labels in rank order. To help overcome this limitation and to clarify the occurrences seen, Table 2.2

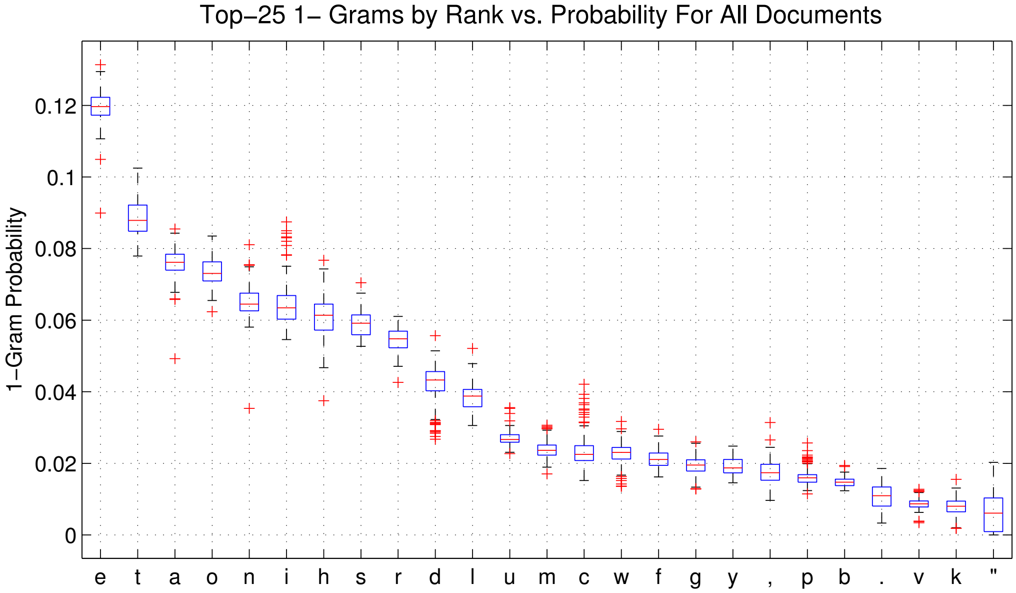


Figure 2.7. For each document in the example data set, the charts show the rank index of each n-gram on X and the corresponding probability on Y where A through C corresponds to n of 1 through 3.

shows the top 10 n-grams for each n. Earlier we saw that the top three(3) words in the example data are 'the', 'and', and 'of'. Perhaps not surprisingly, we see all the characters these words except 'f' in the top 10 1-gram list. However, only the word 'the' is fully captured by the top 10 2-grams. Further, all of the top three(3) words may be reconstructed using the top 10 3-grams. The important thing to take away from these observations is that differing values for n capture subtle differences in occurrence from each other. The value of these specific differences varies by application.

The hyperbolic distributions observed in n-gram frequencies has a significant impact on entropy measurable for these types of systems. Table 2.3 shows dimensions of the example data set for n-grams 1 through 3. We recall that the maximum possible entropy for any probability distribution of size X is $\log_2(X)$. Thus the table shows the following:

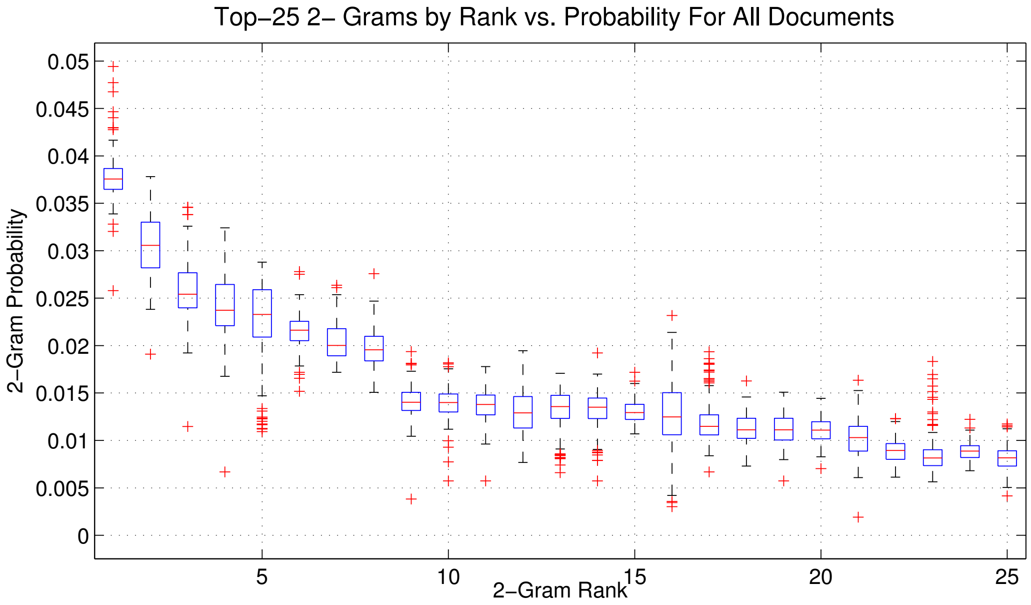


Figure 2.8. For each document in the example data set, the charts show the rank index of each n-gram on X and the corresponding probability on Y where A through C corresponds to n of 1 through 3.

1. The theoretical maximum number of n-grams $||\mathbb{G}_n||$
2. The consequent maximum entropy for that theoretical size $H_{max} = \log_2(||\mathbb{G}_n||)$
3. The actual number of n-grams $||G_n||$
4. The consequent maximum entropy for that actual size $\log_2(||G_n||)$
5. The calculated entropy for G_n where $H(P(G_n)) = -\sum P \log_2(P)$
6. And finally, the ratio of $H(P(G_n))/H_{max}$

For each type of n-gram, we see a loss of entropy due both to the smaller actual size of unique n-grams seen, as well as the hyperbolic distribution that is far from the maximal entropy expected from a uniform distribution. Furthermore, while the actual entropy H increases with n, the ratio of $H(P(G_n))/H_{max}$ decreases with n. The intuition from this observation is that, the larger the n, the greater the information

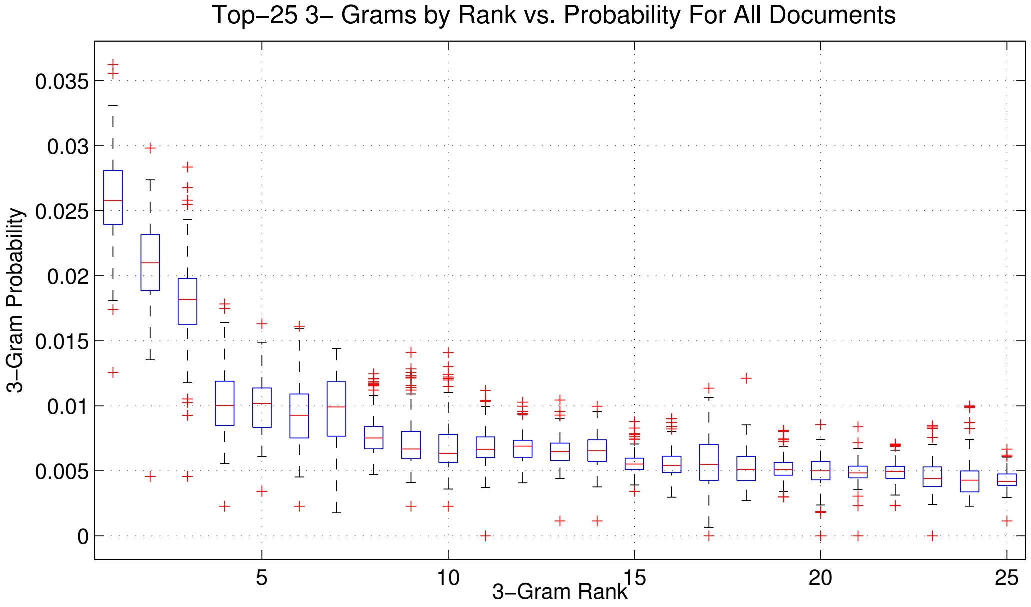


Figure 2.9. For each document in the example data set, the charts show the rank index of each n-gram on X and the corresponding probability on Y where A through C corresponds to n of 1 through 3.

captured in the probability distribution, and thus the less "surprise" found overall in the data as reflected in the lower entropy $H(P(G_n))$.

Moving beyond the summary aspects of n-grams as reflected in their class of distribution, we look next at how the specific distribution of individual or groups of sequences varies both from the average and among instances.

Each source of data in the *corpus* (i.e., $s \in S$) has its own probability distribution. Figures 2.7 through 2.9 show the combined distributions of the individual documents in our test example superimposed in one chart for each value of n. In these charts, we can see obvious clustering about the mean for each n-gram with occasional "outliers" standing out from the rest of the samples.

We now zero in on a few specific documents from the example data set in Figures 2.10 through 2.11. Each figure shows the 1-gram through 3-gram probabilities

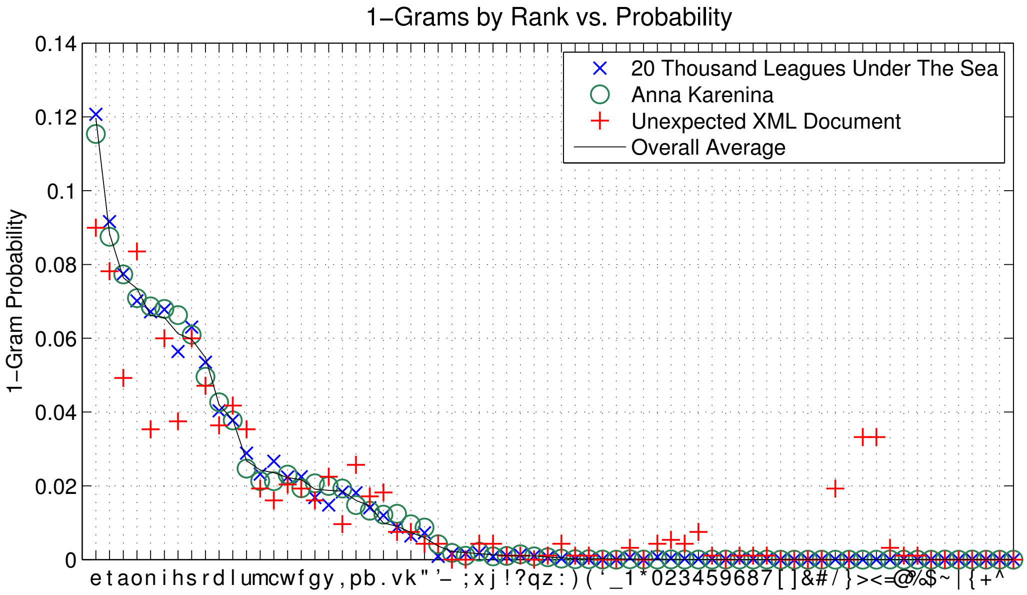


Figure 2.10. The chart shows the rank index of each 1-gram in rank order on X and the corresponding probability on Y for 2 normal documents plus an unexpected XML document as an “outlier,” along with a reference line showing the overall averages for the data set.

for two selected works – Jules Verne’s “20 Thousand Leagues Under The Sea” and Leo Tolstoy’s “Anna Karenina” (English translation) – plus an XML document that unexpectedly found its way into the mix.

Not unexpectedly, the n-gram frequencies of the XML document are far from the mean for many instances at all levels of n. Techniques for detecting anomalous data sources using n-grams should easily be able to recognize such an outlier using only the most basic 1-gram.

On the other hand, the two valid documents have very similar distributions at the 1-gram level, only showing differences for a few specific n-gram instances at the higher levels. For example, several of the 2-gram and 3-gram instances for “Anna Karenina” fall farther from the mean than do the corresponding results for “20 Thou-

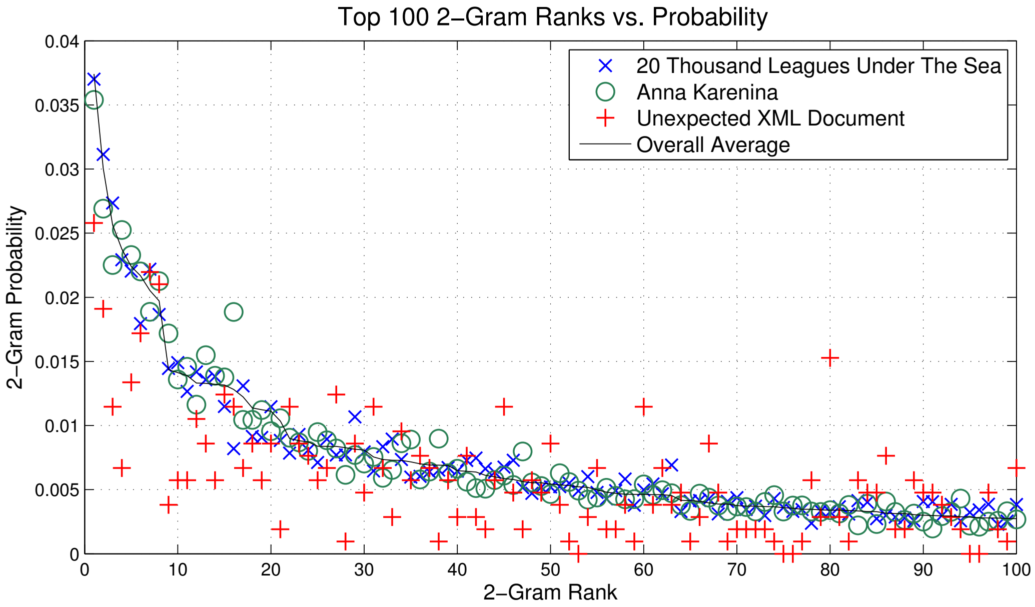


Figure 2.11. The chart shows the rank index of the top 100 2-grams in rank order on X and the corresponding probability on Y for 2 normal documents plus an unexpected XML document as an "outlier," along with a reference line showing the overall averages for the data set.

sand Leagues Under The Sea." We will see later that these more subtle differences at higher levels of n may be used to recognize important differences among data sources not "visible" at the 1-gram level.

2.3.2 Motifs

The study of "motifs" within time series data has received much attention in recent years. A motif is basically some sequence of data that has important qualities to the researcher. For example, a motif may represent some sought-after event or a pattern that is found to occur frequently.

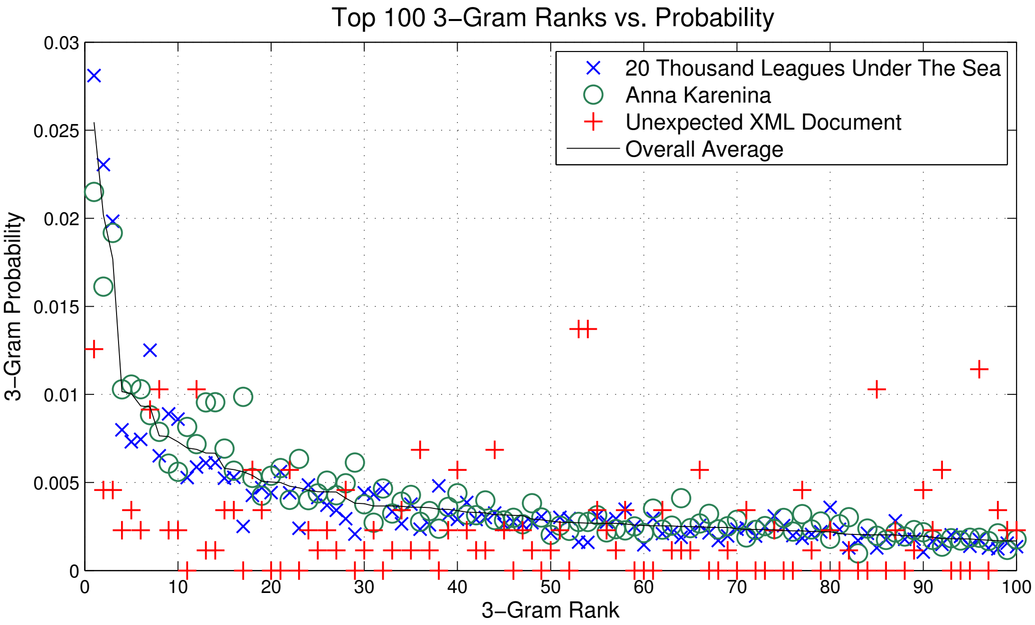


Figure 2.12. The chart shows the rank index of the top 100 3-grams in rank order on X and the corresponding probability on Y for 2 normal documents plus an unexpected XML document as an "outlier," along with a reference line showing the overall averages for the data set.

Some of the earliest research using the motif concept started in the field of genetics where the data sources are the naturally-symbolic sequences of amino acids in gene sequences [42, 43].

More recently, the study of motifs has shifted into the realm of signal processing where time-series data is represented more often as relatively high-resolution samples of some continuous-value data source [44, 45, 46]. In such areas, one of the key challenges is in finding relevant motifs where the start and end of the sequence is uncertain [43, 44, 47], and/or where the data values of similar motifs require some flexible matching scheme, such as the use of Dynamic Time Warping, or DTW [46].

Many areas of study seek to transform the highly-varying data into a more symbolic form through discretization, with the goal of having a more tractable data set to analyze using string processing techniques and the like [48, 49].

As it relates to our later research, motifs in the form of discrete symbol sequences share common characteristics with several of our target data sources. One important similarity that we shall see is that message arrivals occur at irregular intervals, making the starting and stopping conditions of a motif unclear. In later chapters we will discuss ways to address this uncertainty.

2.3.3 Clustering and k-Means

Clustering refers to many types of algorithms and techniques that attempt to identify groups, or "clusters," of data according to some criteria such that members of the group are more alike than members of any other group. Several clustering approaches figure prominently in the literature, including Expectation Maximization (EM) [50] and k-means [51].

For our purposes, we will use k-means as way to identify related groups of data so that we may be able to further analyze their properties. We choose k-means somewhat arbitrarily on the basis of its much-studied use within related research and with the assumption that the specific qualities of any particular clustering approach will help us achieve our goals.

k-means starts with the assertion that a data set may be grouped into k clusters. The method does not specify how k is chosen or whether k is the "right" number of clusters into which the data set naturally groups. But given some value of k and a set of n observations $S = \{s_1 \ldots s_n\}$, the method seeks to find the set of clusters $\mathbb{G} = \{G_1 \ldots G_k\}$ where each G_i has a mean of μ_i such that the "within-cluster sum of squares" (WCSS) is minimized as follows:

$$\arg\min_{\mathbb{G}} \sum_{i=1}^{k} \sum_{s_j \in G_i} ||s_j - \mu_i||^2 \tag{2.12}$$

It has been established that k-means falls into the class of NP-hard problems [52], so it is difficult to find the optimal solution; however, there are several strategies for usable results.

The most common approach uses an iterative approach similar to the EM algorithm. It starts by choose k observations in S that are considered the mean of each set $\mu_1^{(0)}$ through $\mu_k^{(0)}$. There are several ways these k observations can be chosen, but most often, they are chosen at random. Then each observation of S is assigned to an initial set $G_1^{(1)}$ through $G_k^{(1)}$ based on having the smallest Euclidean distance to the set's mean. The resulting sets have their own means $\mu_1^{(1)}$ through $\mu_k^{(1)}$ that likely differ from the initial ones.

This process of assigning observations to sets and computing the new set means is continued until the means before and after converge, or until some maximum number of steps at which point the there is a failure to converge, and the process is tried again with new initial observations. The general form of these steps are as follows for steps $t = 1 \ldots t_{max}$ and given $i, j = 1 \ldots k$, $s \in S$, and some minimum measure of convergence ϵ:

$$G_i^{(t)} = \left\{ s \ : \ \begin{matrix} ||s - \mu_i^{(t-1)}||^2 < ||s - \mu_j^{(t-1)}||^2 \text{ and } i \neq j \\ ||s - \mu_i^{(t-1)}||^2 = ||s - \mu_j^{(t-1)}||^2 \text{ and } i < j \end{matrix} \right\} \tag{2.13}$$

$$\mu_i^{(t)} = \frac{1}{||G_i^{(t)}||} \sum_{s_g \in G_i^{(t)}} s_g \tag{2.14}$$

$$\mathit{converged?} = |\mu_i^{(t)} - \mu_i^{(t-1)}| \leq \epsilon \tag{2.15}$$

Note that the means in this case may, in fact, be vectors, and as such represent a location in some multi-dimensional space encompassing the observations of S. As such, it is common to call them "centroids" for the clusters. A convenient property of these centroids that we will exploit later is that, once chosen, possible "new" observations for some expanded data set S' can be assigned to one of the $\mathbb{G}$ clusters using the conditions specified in Equation 2.13.

As stated earlier, k must be chosen prior to the application of k-means to find clusters within some data set. How to choose k depends on the problem domain, and we shall see later a method to test multiple choices for k, selecting the "best" choice.

However, in practical terms, data sets whose observations are "near" each other in Euclidean terms, tend to fall within the same cluster regardless of the value of k. For example, in our data set of classic literature documents, we have 11 "Tom Swift" novels by Victor Appleton. In experiments applying k-means to the n-gram frequency distribution of these documents using a range of choices for k, all 11 "Tom Swift" novels were together in one of the resulting clusters. This outcome is perhaps unsurprising given the expectation from prior studies that the works of a given author tend to have similar n-gram distributions, and that as such, their distances from any cluster centroid will likely be similar, resulting in similar cluster assignments.

2.3.4 Finding The "Best" k

Many methods have been studied for finding optimal values of k [53, 54, 55, 56]. One typical approach is based on minimizing the sum-of-squares within clusters (SSW) – sometimes called the "cluster distortion" – and maximizing the sum-of-squares between clusters (SSB) as presented by [57]. The method is defined using the following equations given $i = 1 \ldots k$, $\bar{X}$ is the mean of the entire data set, C_i is the centroid of a cluster, and G_i is the elements assigned to the cluster.

$$SSW(k) = \sum_{i=1}^{k} \sum_{g \in G_i} ||g - C_i||^2 \tag{2.16}$$

$$SSB(k) = \sum_{i=1}^{k} ||G_i|| \times ||C_i - \bar{X}||^2 \tag{2.17}$$

$$J(k) = k\frac{SSW(k)}{SSB(k)} \tag{2.18}$$

By performing clustering for a range of values for k, each producing a set of cluster assignments and centroids, then computing the value of J for each result, we will select the value of k which has the smallest value of J.

2.3.5 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most widely-used tools for analyzing data represented in matrix form. PCA has been heavily studied and the mathematics behind it have many formulations.

The basic concepts behind PCA are straightforward, but the applications are varied and sometimes surprising. Given some n-dimensional data set having m observations, PCA will create a square $n \times n$ "coefficients" matrix with special properties. This matrix contains an ordered set of *eigenvectors* that can transform the original data set into a new $n \times m$ data set, sometimes called "scores," whose new dimensions capture the greatest to least amount of covariance within the original observations.

To illustrate how the coefficients matrix is derived and how it may be used, we return to our example data set of text documents, and 1-grams in particular. Figure 2.13 plots the various 1-gram probabilities for each of the documents in our data set with the 1-gram rank on X and the probabilities on Y. Our intent is to show that range of probability values that may be found for each 1-gram in general, but we highlight the "xml outlier" to be explained later.

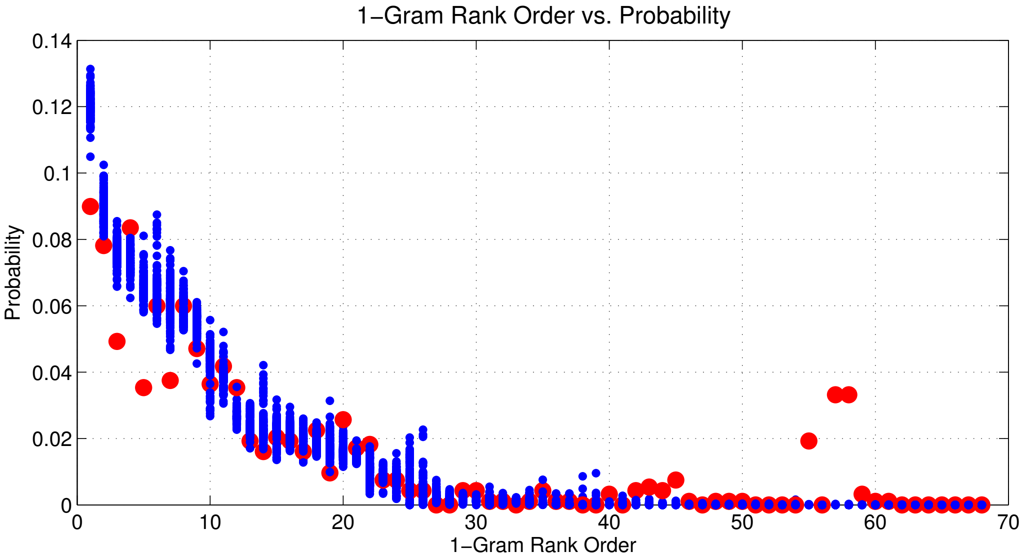


Figure 2.13. The chart shows 1-gram rank order on X and plots the probabilities for each document in the example data set on Y to illustrate the variability of values for each rank, with special emphasis given to the “xml outlier” using larger red dots.

PCA starts by requiring all the data in each dimension – in our case, the dimensions are the 1-grams – to be “mean adjusted;” that is, to offset the data values in each column such that mean is at zero and all values have a positive or negative value based on their distance from the mean. For our example, this is shown in Figure 2.14. We note that the range of values in each dimension can be large even as the rank of the 1-gram decreases. Also, the “xml outlier” has both values far less than the mean for higher-rank 1-grams and values far higher than the mean for very low-rank 1-grams.

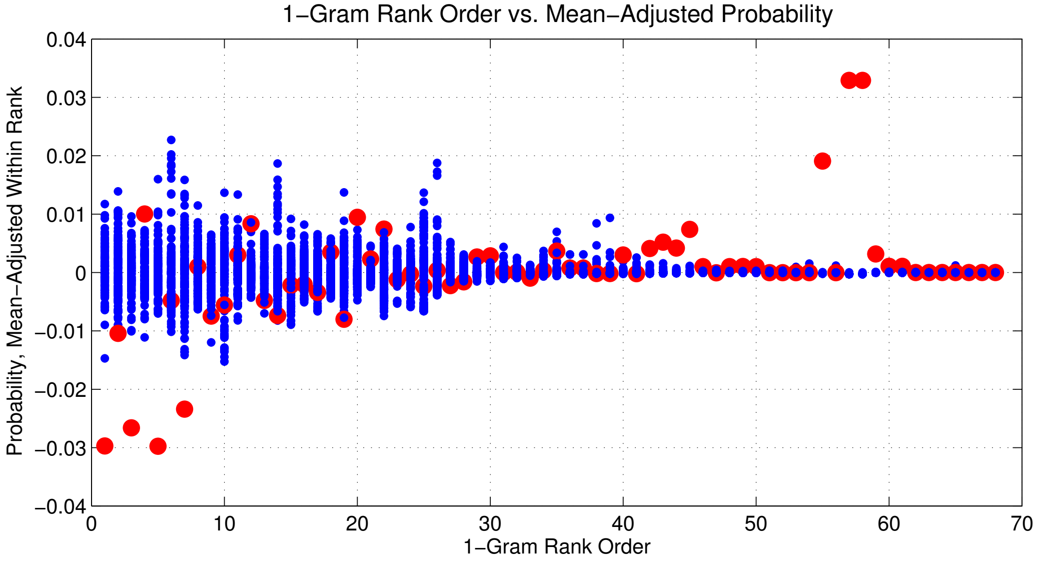


Figure 2.14. The chart shows 1-gram rank order on X and plots the mean-adjusted probabilities within each rank for each document in the example data set on Y to highlight the variability of of values regardless of rank, with special emphasis given to the “xml outlier” using larger red dots.

$$\bar{X} = \left\{ \bar{x_j} : \frac{1}{m} \sum_{i=1}^{m} x_{i,j} \;\forall j = 1 \ldots n \right\} \tag{2.19}$$

$$Y = \{ y_{i,j} : x_{i,j} - \bar{x_j} \;\forall i = 1 \ldots m \;\; \forall j = 1 \ldots n \} \tag{2.20}$$

$$covX = \frac{1}{m} Y Y^T \tag{2.21}$$

$$D = \{ d_i : covX_{i,i} \;\forall i = 1 \ldots n \} \tag{2.22}$$

$$V^{-1}(covX)V = D \tag{2.23}$$

$$C = \left\{ \begin{matrix} v_{*,i} \text{ if if } d_i > d_j \\ v_{*,j} \text{ if if } d_i <= d_j \end{matrix} \quad \forall i = 1 \ldots n \;\; \forall j = 1 \ldots n \right\} \tag{2.24}$$

Given a matrix X with n dimensional columns and m observational rows, the $n \times n$ coefficients matrix C is constructed by computing the covariance matrix (Equation 2.21) of the mean-adjusted data set (Equation 2.20), computing eigenvectors V

from the covariance matrix (Equation 2.23), and then ordering the eigenvectors in descending order based on their magnitude (Equation 2.24).

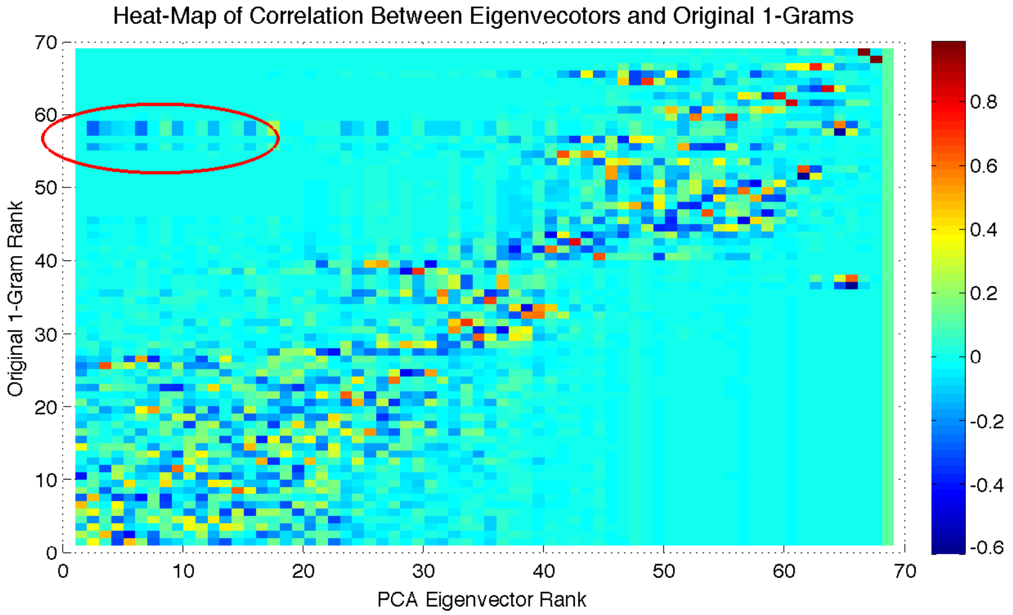


Figure 2.15. The heat map chart shows the correlation between PCA eigenvector rank on X against original 1-gram rank on Y; a high positive intensity in red indicates high correlation; a high negative intensity in dark blue shows a high anti-correlation, with no correlation showing as a neutral light-blue; in addition, the strong presence of XML tag characters for the "xml outlier" are circled in red.

The contents of the coefficients matrix for our data set can be seen in the form of a "heat map" in Figure 2.15. Each column of in the matrix represents the eigenvector for the new coordinate system into which the data may be mapped in descending order. The left-most column represent the axis whose data values have the greatest variance within the entire data set. Each row of the matrix corresponds to the original 1-gram dimensions. By examining this column, we can see which 1-grams were highly correlated and contributed the most to this new "principal axis"

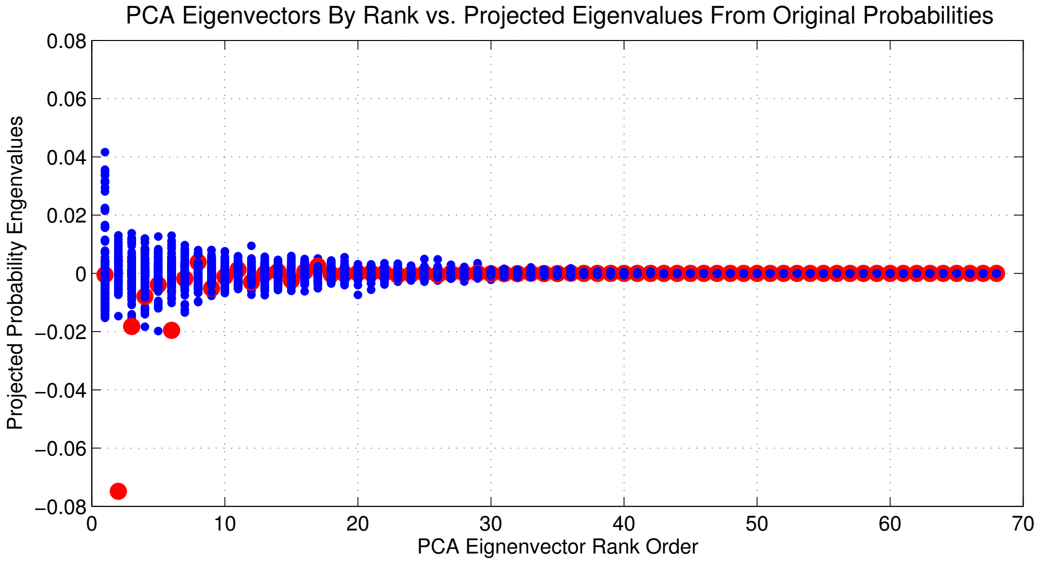


Figure 2.16. The chart shows the rank order of the principal axes resulting from PCA on X and plots the "scored" probability values from the original data set on Y, with special emphasis given to the "xml outlier" using larger red dots.

by finding those with the largest positive magnitudes. Those rows with the largest negative values represent values that represent anti-correlations on this axis.

Looking closely at this heat map, we can see that several, but not all, of the high-ranking 1-gram dimensions are correlated and have strong participation in several of the left-most principle axes. We can also easily see the effect of a strong anti-correlation due to the XML tag characters in the top-left area, representing their low-rank in the original data set and negative contribution to many of the most significant, and therefore left-most, axes.

As stated, with a coefficients matrix, we are able to define a new set of values E that map the original data into the *eigenspace* whose dimensions are governed by the eigenvectors of the matrix.

$$E = XC \tag{2.25}$$

Figure 2.16 shows the result of passing the original 1-gram data through the coefficients matrix (Equation 2.25) and plotting the result. As compared to Figures 2.13 & 2.14, we note the charts have similar dimensions, but that the X axis is no longer the descending rank order of 1-grams nor is the Y axis a probability. Instead, the X axis is the descending rank order of the principal component axes and the Y axis plots the magnitudes of the "scored" original probabilities within their new dimensional ranges. By showing the values of the "xml outlier" on this chart, we see how the various values from the original data which were often anti-correlated to the data set as a whole, continue to make their presence known as a strong outlier within the second principal axis, as well somewhat less strongly in the third and sixth axes.

This leads us to note another important characteristics of PCA – its ability to preserve statistical significance within a data set while reducing the number of dimensions of the data. When working with high-dimensional data, it is often helpful to reduce the number of dimensions being considered for further analysis. Fewer dimensions can be computationally faster to process and easier to visualize.

In our example of the "xml outlier," we see that the "scored" data captures some of the statistical properties of the low-rank values within the new high-rank principal axes. As a result, the lower-rank principal axes may be safely discarded as providing little statistical significance to the overall data set. By transforming the "scored" data back into the original coordinate system using only the most-significant eigenvectors of the coefficients matrix, we can create a revised original data set that is only based on these most-significant principal components – this is sometimes called "whitening" the data. The following equations show how this can be achieved using the following equations using some number of eigenvectors L.

$$W = \left\{ \begin{array}{ll} v_{*,i} & \text{if } i <= L \\ 0 & \text{if } i > L \end{array} \quad \forall i = 1 \ldots n \;\; \forall j = 1 \ldots n \right\} \tag{2.26}$$

$$X' = XWW^{-1} \tag{2.27}$$

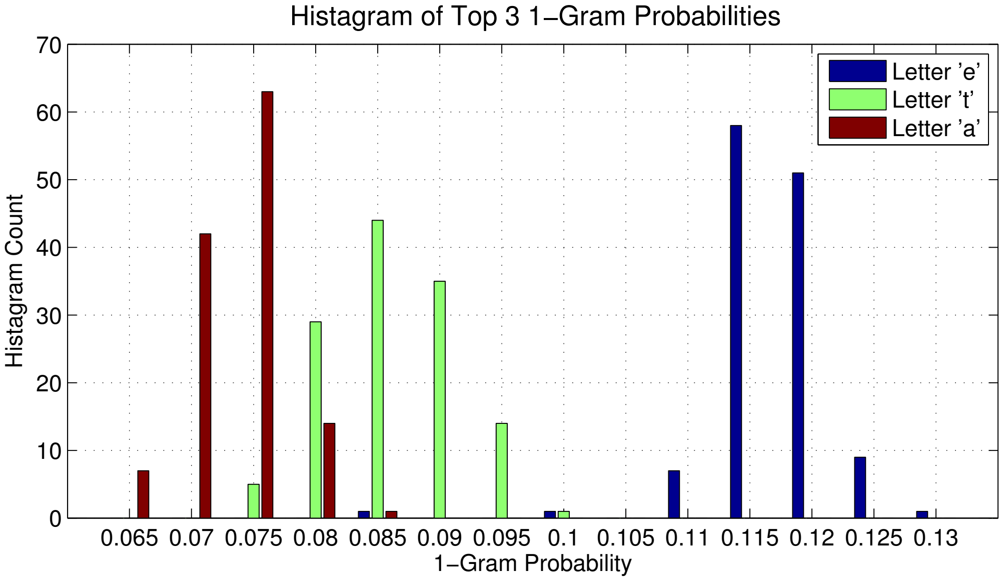


Figure 2.17. The chart shows a histagram of the probability distribution of the top 3 1-grams in the example data set, illustrating their apparent Gaussian distribution.

As seen in Equation 2.21, the heart of PCA is the creation of a covariance matrix that relies effectively on a “least squares” calculation to capture correlations within the data. It is well known that these calculations are sensitive to “outliers” and that it is important to establish that the data being considered follows a Gaussian distribution, as illustrated in Figure 2.17 for our example data set.

Various research efforts have explored the use of PCA, sometimes using different names, as a precursor to applying clustering algorithms such as k-means [58, 59, 60]. The general finding is that the process of transforming a data set from its original coordinate system into one that emphasizes its inherent statistical correlations helps various clustering algorithms, and in particular k-means, to more effectively identify clusters.

CHAPTER 3

RESEARCH FOCUS

3.1 Process Overview

At this stage, we have established the mathematical notation and elaborated on key mechanisms that form the foundation of this dissertation. Furthermore, we have reviewed related research work in these areas that have previously explored the problem space. We now turn our attention to a class of problems that allow us to build on the prior art and reveal new insights into the behavior of systems based on their communications flow.

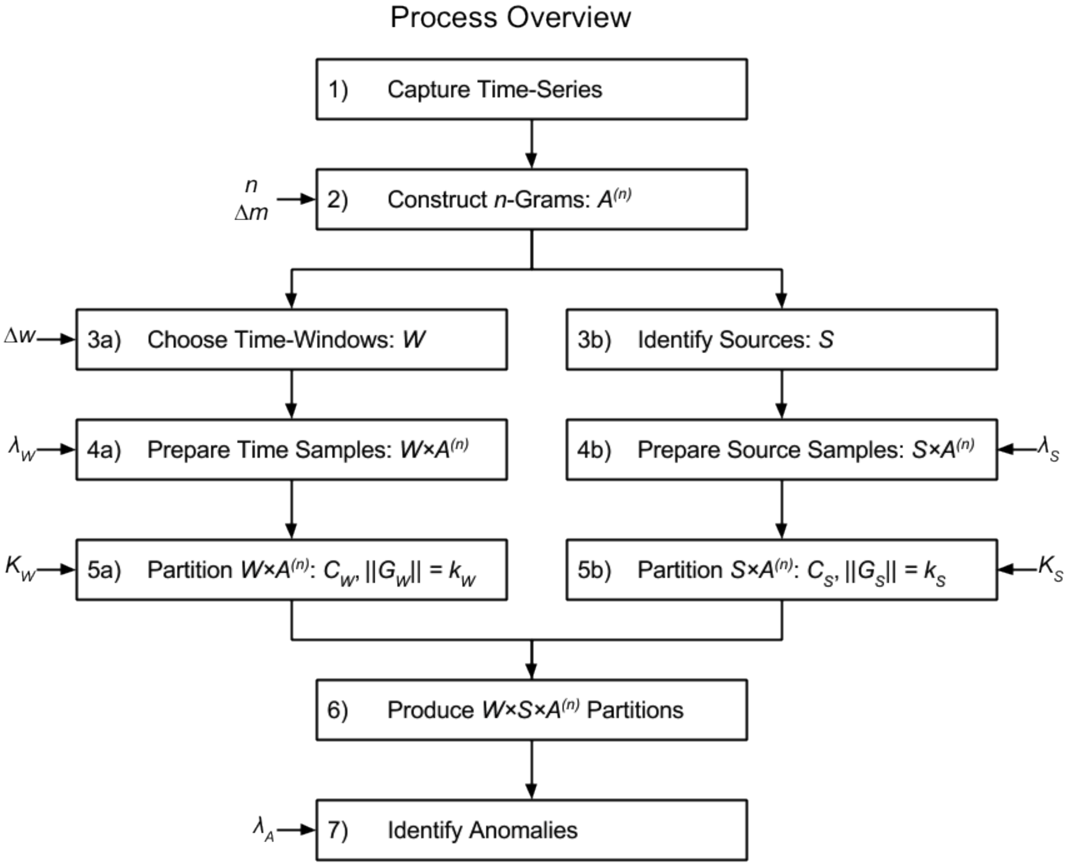
We start with the basic proposition that, for any system with an observable communications flow, we may characterize aspects of its behavior both temporally and demographically using a combination of absolute and relative statistical properties. The outcome of our characterization is a collection of statistical intersections that can be expected to occur over time.

As new observations for a system arrive, we can categorized them as matching one or more of the known intersections, or in failing to do so, identify them as anomalies requiring further investigation. Once identified, the expected intersections and unexpected anomalies may be assigned some meaning, depending on the system being observed.

As anomalies occur, we also have the opportunity to review and revise the characterization intersections to account for their presence.

In order to characterize some target system, we must ask the following questions: 1) Who was "talking"? 2) When were they "talking?" 3) What were they "saying?"

We can answer these questions for a given temporal message domain $\mathbb{M}$ having a set of functions $\mathbb{V}$ to extract information using our notation. Given a subset of messages $M \subset \mathbb{M}$, we have a function $\mathcal{V}_S(M) = S$ for the "sources" of the messages, answering question #1. We can answer question #2 with the function $\mathcal{V}_T(M) = T$ for all messages, and $\mathcal{V}_T(\mu) = t$ for any specific message.



Process Overview
1) Capture Time-Series
n
Δm
2) Construct n-Grams: $A^{(n)}$
Δw
3a) Choose Time-Windows: W
3b) Identify Sources: S
λ_W
4a) Prepare Time Samples: $W \times A^{(n)}$
4b) Prepare Source Samples: $S \times A^{(n)}$
λ_S
K_W
5a) Partition $W \times A^{(n)}$: C_W, $\|G_W\| = k_W$
5b) Partition $S \times A^{(n)}$: C_S, $\|G_S\| = k_S$
K_S
6) Produce $W \times S \times A^{(n)}$ Partitions
λ_A
7) Identify Anomalies

Figure 3.1. The figure shows the high-level process stages from the original acquisition of data through the marginal analyses, culminating in the X-based analysis to identify anomalies.

So far, the answers are pretty obvious, but answering question #3 can be more difficult. For message domains of any complexity, there may be many additional functions available to extract information content in the form of both discrete and continuous values. As a result, we must choose those functions from which we can construct a discrete symbol set that encodes some relevant subset of the information available. There may be multiple options for this encoding similar to the choice of letters or words in textual analysis. For our purposes, we will call this function $\mathcal{V}_A(M) = A$ and consider A to be the "lexicon" or "alphabet" for further analysis.

With this as a foundation, we may apply the analysis process shown in Figure 3.1 and described as follows:

1. **Capture Input Time-Series** – Construct an ordered $3 \times ||M||$ matrix where each row consists of a source s, symbol a, and time t of the form (s, a, t) where $s \in S$, $a \in A$, and $min(T) \leq t \leq max(T)$.
2. **Construct n-Grams** – With A as the set of "1-gram" symbols occurring in M, we may rewrite it as $A^{(1)}$ and we will denote the set of symbols representing additional levels as $A^{(n)}$. So for example, the set of 2-grams are written as $A^{(2)}$. In addition for n-grams where $n > 1$, we may want to limit symbol sequences based on the concept of "motifs" (see Section 3.2).
3. **Establish Source and Time Dimensions** – For sources, this step is simply the materialization of the set S. For time, we choose a time window size Δw that allows us to aggregate occurrences of $A^{(n)}$ into statistical "bins" for further analysis, the set of which we call W (see Section 3.3).
4. **Prepare Source and Time Samples** – With each of the elements of either S or W represented as sample rows and elements of $A^{(n)}$ as the columns, we construct two-dimensional matrices $S \times A^{(n)}$ and $W \times A^{(n)}$, respectively. We

may then apply various additional transformations on these matrices in order to ready them for the next stage in the process (see Section 3.4).

5. **Partition Source and Time Samples** – Given $S \times A^{(n)}$ and $W \times A^{(n)}$, we next perform multiple evaluations of the k-means algorithm to find the "best k" that partitions S and W into a set of groupings. We refer to the source-based groupings as Γ and the time-based groupings as Φ, each having its own "best k" size such that $||\Gamma|| = k_\Gamma$ and $||\Phi|| = k_\Phi$, respectively. Each element of S and W is assigned membership to $\gamma_i \in \Gamma$ where $i = \{1 \cdots k_\Gamma\}$ and $\phi_j \in \Phi$ where $j = \{1 \cdots k_\Phi\}$, respectively as well (see Section 3.5).
6. **Intersect Source and Time Partitions** – Using the source- and time-base partitions of Γ and Φ, construct a combined three-dimensional partitioning of M into $S \times W \times A^{(n)}$ that we call X (see Section 3.6).
7. **Analyze Source, Time, and Cross-Section Partitions** – At this point we are able to perform various statistical analyses on the partitions Γ, Φ, and X to both characterize M in various ways and to identify anomalous samples that fall outside of the norms established by these partitions (see Section 3.7)

When applying the process, we first start with some captured history defining W. But once we have produced our partitions – Γ, Φ, and X – we have two basic choices for dealing with new data for times where $t_{new} > max(T)$.

- **Baseline Processing** – The first and simplest approach is to use the $\Gamma \times \Phi \times X$ partitions as a static characterization of the system, comparing each subsequent time window of size Δw to find which time window partition ϕ. Since each of the sources have a definite assignment to a source partition γ, once the time window is assigned to a ϕ, we also have a χ assignment as well. The advantage of this approach is that only the data for a single time window need be considered for progressive analysis, but it cannot adapt to a changing "normal" if the current state of the system differs from the past and we want to consider this to be the "new normal."

- **Sliding Window Processing** – To address the problem of a changing definition of "normal," we can shift the time range over which we define W and redefine the $\Gamma \times \Phi \times X$ partitions accordingly. Of course, the reprocessing of the data set to produce the partitioning may also be quite expensive and may not be something that is practical for each newly available time window.

3.2 Defining A

What we choose as a symbol set A extracted from the possibilities within the message domain $\mathbb{M}$ depends greatly on the domain itself. But we must have knowledge of the domain perhaps to decode the data format to extract one or more discrete values whose combinations can be assigned to individual symbols, it does *not* follow that we must know the significance of the values or have any preconceived notions of their merits – the data itself will tell us that as a consequence of the analytical process.

Whatever the initial symbol set chosen, we may assume that symbol arrival within messages for the target system conveys some information, and effectively form a "language" used by the message sources.

Therefore, just as we saw in our explanation of n-grams earlier, we may reasonably expect that measurable information content of those sources based on such a symbol set – the "1-grams" – may be enhanced by considering sequence combinations as well in the form 2-grams, 3-grams, and the like.

Since we are collecting statistics within discrete time window boundaries, it may easily occur that any two symbol occurrences in the channel may straddle such a boundary. As a result, we must decide how to count any n-gram occurrence in such cases. As a convention throughout the remainder of this dissertation, we will assume that n-grams frequency will be counted in the time period within within the terminating symbol occurred.

As we will see in some of our studies, messages from some sources may arrive in bursts, but with no definitive boundary separating such bursts. As a consequence, we may have widely varying time delays between some symbol pair occurrences. The question then becomes: Should we count n-grams regardless of the time interval between symbol occurrences?

While we will explore options for maximum relevant time intervals within our subsequent experiments, we will assert in principle that there *is* a period after which an $n - gram$ will not be counted.

In addition to n-grams at the "alphabet" level, we may also consider treating message "bursts" some what like a "dictionary" of words as we saw earlier with motifs. We may even further apply an n-gram approach to counting such motif sequences.

Whether a simple "alphabet" of per-message symbols, a "dictionary" of message motifs, or n-gram sequences of either, the resulting discrete, unique occurrences may be used to define the final A used for the subsequent analysis using our process.

However, as we know from our introduction to n-grams and the nature of hyperbolic probability distributions, there may be a large number of symbols in A which rarely occur and offer little statistical impact.

Rather than discard these symbols, we choose to truncate A by eliminating the least-occurring symbols, replacing them with an "other" symbol to which the truncated symbols are mapped. Ideally, the probability of the "other" symbol is such that it is less than or equal to that of the least-significant remaining symbol in A. The resulting symbol set we call A' and the corresponding function on M is therefore properly defined as $\mathcal{V}_{A'}$.

Using our notation, we now define two matrices representing the absolute number of occurrences of symbols in A' by time period and by source with Equations 3.3 and 3.4. Each has columns corresponding to the symbols in A', where the row of V_W contains the occurrences of those symbols in separate time periods of size W, and where each row of V_S contains the occurrences of those symbols for each source.

As suggested earlier, an examination of the overall distribution of the symbols of A for a collection of messages M, and more generally, for the overall message domain $\mathbb{M}$, often follow Zipf's Law; that is, having a hyperbolic distribution. We can easily compute the overall distribution of A using Equation 3.1. This distribution has a descending rank order that we can express as $\rho(A)$, where $\rho(a_i) = 1$ when a_i has the highest probability in A, and $\rho(a_j) = ||A||$ when a_j has the lowest probability in A. With $P(A)$ and $\rho(A)$ and some target size of $||A'|| = n$, the definition of A' is defined in Equation 3.2.

$$P(A) = \frac{||M_A||}{||M||} \tag{3.1}$$

$$A' = \{\forall a \in A \mid \text{if} \rho(a) < n \text{ then } a, \text{ otherwise `}\textit{other}\text{' } \} \tag{3.2}$$

While using A' in practice, we may, for convenience simply use A elsewhere in our notation unless we are explicitly discussing the differences between the two sets.

3.3 Choosing W

When choosing W, we must balance the need for the sufficient statistical significance of the resulting aggregation periods with the need for useful resolution of the intersect space in the temporal dimension. Depending on the message domain, there may be multiple choices possible, each with its own justification. Small values for W

provide high resolution over the temporal dimension, and paired with a larger value of k, could help make fine distinctions among the intersects possible for the system.

3.4 Choosing Conditioning Options

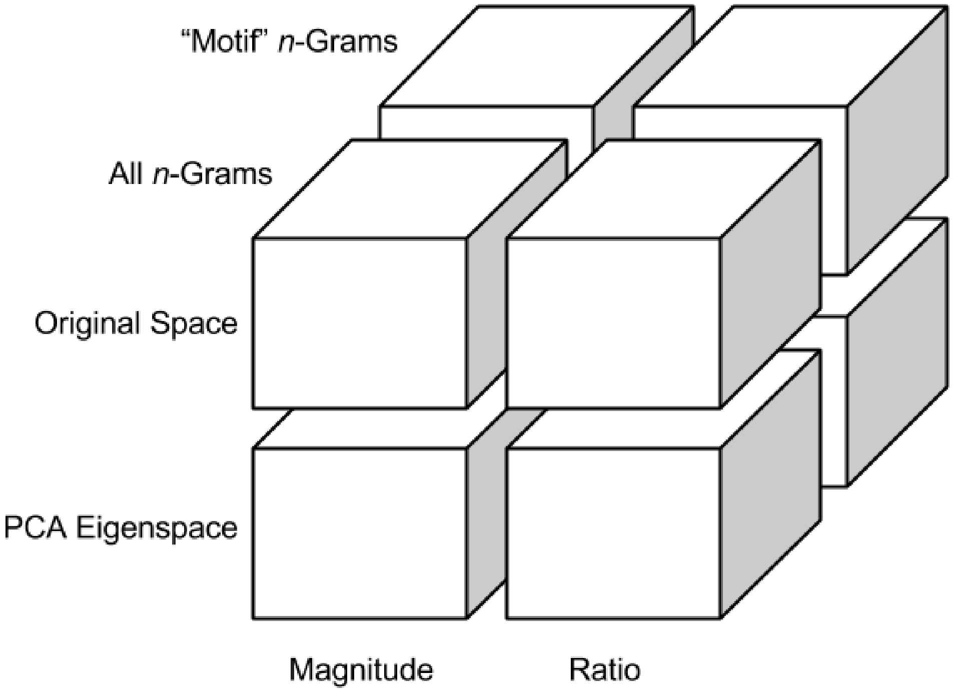


Figure 3.2. This figure illustrates the combinations of parameter choices when conditioning data in preparation for the partitioning process; each box represents a conditioning scenario.

With established definitions for S, W, and $A^{(n)}$, we are able to construct two parallel two-dimensional matrices with S and W as the rows and $A^{(n)}$ as the columns, which we are $S \times A^{(n)}$ and $W \times A^{(n)}$, respectively. After constructing these matrices, we have many options for further conditioning of the data to prepare it for the subsequent stages of the process as illustrated in Figure 3.2.

In Section 3.2, we addressed whether or not to limit n-grams by the notion of "motif," but additionally, we may choose to work with the original occurrence counts of $A^{(n)}$ within each sample for S and W, or we may convert these values into probability distributions of their occurrence within each sample. Additionally, we may use PCA as a tool to help identify similarities among the samples.

$$V_W = ||M_A^W|| \tag{3.3}$$

$$V_S = ||M_A^S|| \tag{3.4}$$

Using V_W and V_S to capture the scale of M, we may also define two additional matrices to capture the lexical "mix," or ratios, of those occurrences by row with Equations 3.5 and 3.6.

$$R_W = \frac{V^W}{||M_A||} \tag{3.5}$$

$$R_S = \frac{V^S}{||M_A||} \tag{3.6}$$

With matrices for magnitude V and ratio R, we are ready to classify M into window and source types using PCA and k-Means. As we saw in the introduction, PCA is able to identify relationships among the dimensions of a data set – in our case, the occurrences of A in M. Thus we pass our matrices through PCA to construct the *eigenvectors* defining a new space that emphasizes those relationships. Next we project our data into the new *eigenspace* and then apply k-Means to assign the time periods of W and sources S to window types Φ and source types Γ, respectively. The actual number of window and source types is defined by the size of k for each marginal classification by W and S. We use k_Φ and k_Γ for the number of window and source types, respectively.

As mentioned above, we may classify the time periods and sources of M by either “scale” or lexical “mix.” “Scale” effectively considers the count of messages, while “mix” considers the ratio, or probability distribution, of the “alphabet” used to communicate. Each perspective has its own ability to characterize M. For example, we expect changes in the message occurrence rate throughout the day and from day-to-day for systems that exhibit a diurnal cycle of activity. However, even though the amount of information may change throughout the day, the probability distribution of the communications may remain constant. On the other hand, if a system exhibits constant communications flow with little change in volume over time, the content of the messages may change in their distribution, representing other important changes in activity that warrant separate classification.

3.5 Choosing Source/Time Partitioning Options

The choices of k_Φ and k_Γ are somewhat arbitrary. Small values limit the “resolution” of intersects, while larger ones may create unnecessary computational or storage overhead and may provide little additional value. The key to “good” values of k is their ability to partition the available data into clusters that have a minimum of internal variation and are maximum of separation among each other. Since we would rather learn, rather than impose, the “intersect resolution” of our system, we need a way to automatically choose our values of k.

Since k-means is not guaranteed to produce an optimal result for any evaluation of a given data set, it is necessary to perform multiple evaluations for each value of k and test each for how well it meets the criteria. We saw in Section 2.3.4 that multiple approaches exist for making this determination, but for our purposes which we will detail in our experiments in Chapter 4, we will settle on a simplified method simply using the SSW function defined in Equation 2.16 and choose the evaluation results where this value is minimized.

By finding the "best" evaluation for each k for some range where $k \geq 2$ but also $k \leq k_{max}$ for some pre-selected value of k_{max}, we must next decide which value of k to use. To do this, we simply select the value of k with the lowest overall SSW or "cluster distortion" value. However, as k increases the time complexity also increases proportionally, so for practical reasons we want to find a balance between $SSW(k)$ and k. For this reason, we use a cost function J in Equation 3.7 and find the "best k" where $argmin_k J(k)$ with choices for constants C_1 and C_2 provide weighting factors to help tune the equation based on experimentation.

$$J(k) = C_1 SSW(k) + C_2 k \tag{3.7}$$

A variant of k-means with some unique characteristics that we will find useful later is called "kernel" k-means, or KK-means. As presented by [61, 62], KK-means allows us to replace the Euclidean distance "kernel" function for determining membership in a cluster with some other function that computes a "distance" between two samples using some other means. [61, 62] use this approach to introduce weighting factors to the terms of each sample, but once weightings have been applied, the fundamental calculation remains a minimization of the sum of squares.

$$\arg\min_{\mathbb{G}} \sum_{i=1}^{k} \sum_{s_j \in G_i} KLD(s_j, \mu_i) \tag{3.8}$$

As a unique contribution of this dissertation, we consider using KL-Divergence as an alternative distance function for those data sets whose samples are intrinsically probability distributions of the symbol occurrence ratios within each sample. The challenge in doing so is that KL-Divergence does not obey the triangle property of a true "metric" and when introduced into the k-means algorithm can easily fail to converge. As a consequence, our algorithm must use a heuristic for determining that successive iterations are failing to converge and terminate the evaluation. Since we are already performing multiple evaluations for each value of k as mentioned earlier, failed evaluations are simply discarded. And in some situations, if we fail to achieve any converged results for a given k, we simply discard the k altogether from consideration for the "best k."

3.6 Computing Source/Time Intersections

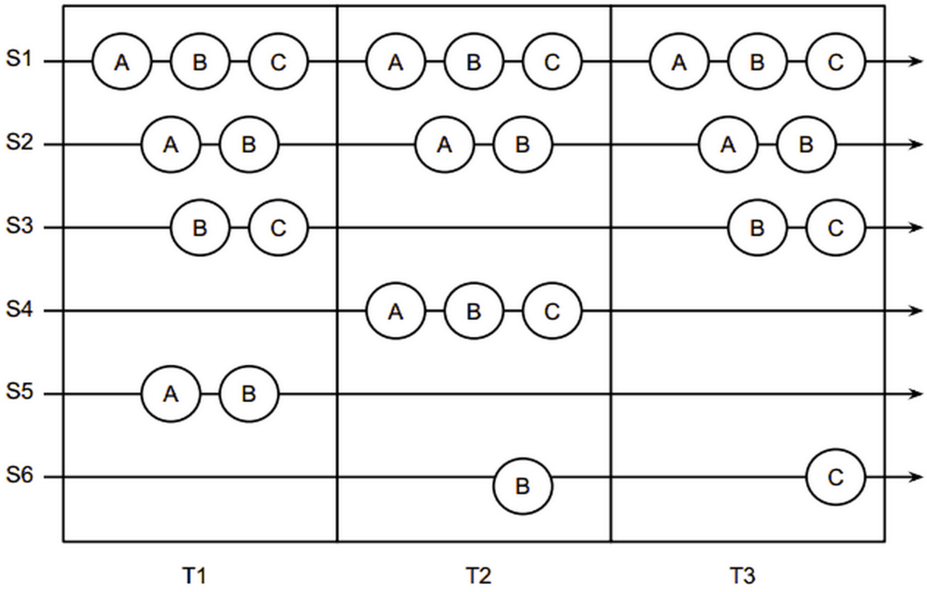


Figure 3.3. An illustration of multiple message sources – $S1$ through $S6$ – communicating one of several symbols – A, B, C – over a series of time intervals – $T1$ through $T3$.

For a more concrete example, consider Figure 3.3 which shows six different message sources labelled $S1$ through $S6$. It should be easy to see that the volume of messages decreases from $S1$ through $S6$. I should also be easy to see that sources

$S1$ and $S4$ repeat the sequence ABC, while $S2$ and $S5$ repeat the sequence AB, and finally, $S3$ and $S6$ repeat the sequence BC. If we choose $k_\Gamma = 3$, we might find the following source types by absolute count – $(S1, S2)$, $(S3, S4)$, and $(S5, S6)$ – and different source types by "mix" – $(S1, S4)$, $(S2, S5)$, and $(S3, S6)$. In this example, $T1$ through $T3$ contain a similar total count of messages and might easily be classified together in the same window type.

Having established marginal classifications for M by time window type Φ and population source type Γ for both volume and ratio of occurrences of symbols in A, we use each combination of window type ϕ and source type γ to create statistical "intersect" $\chi = (\phi, \gamma)$. This gives us two sets of intersects, one based on volume called X_V and another on ratio called X_R. Each intersect defines a subset of occurrences of A that has a statistical *signature*. For intersects based on volume, we have mean, standard deviation, minimum, and maximum values as statistics for each symbol in each intersect. Whereas for intersects based on X_R, we have an average probability distribution $P(A)$ and a KLD for each of the samples making up X_R with respect to that average. These KLD values for a "half-gaussian" distribution having standard deviation.

As illustrated in Figure 3.4, we can think about M as a three-dimensional space where S, W, and A bound a volume. Using Γ, Φ, and A, M may be partitioned into many non-overlapping cuboid regions such that the entire $S \times W \times A$ space is contained in one and only one cuboid corresponding to each $\chi \in X$ – the gray portion of the figure showing one such cuboid.

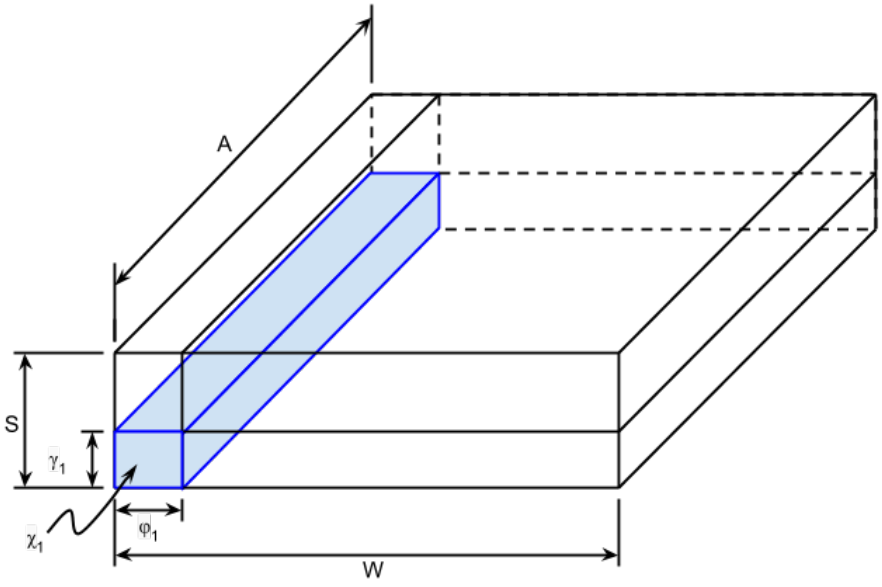


Figure 3.4. The figure shows M as a volume of three dimensions S, W, and A along with a cuboid in light blue showing χ_1 representing a portion of M further bounded by γ_1 on the S axis, ϕ_1 on the W axis.

3.7 Detection of Anomalies

Once we have partitioned our data into window types Φ, source types Γ, and intersects X, we are now in a better position to look for anomalies within these subsets. For convenience, we can classify anomalies along several lines as follows:

- **Source Anomaly** – The statistical properties of a given source within M falls outside established thresholds for any source type in Γ.
- **Window Anomaly** – The statistical properties for a given time period within M falls outside established thresholds for any window type in Φ.
- **Intersect Anomaly** – The statistical properties for the sources of a source type γ for a time period assigned to window type ϕ from M falls outside established thresholds for the expected intersect $\chi = (\phi, \gamma)$.

- **Migration Anomaly** – Since our assignment of samples to “intesects” in X are based on membership in Φ and Γ, it is possible that an an “intersect anomaly” may in fact be a *good fit* for membership in another intersect than the one to which it has been assigned. In a sense, we can think of the sources within a time window as defined by χ as “migrating” to match another χ.

How we define the notion of falling “outside of established thresholds” when identifying anomalies depends on the statistical characteristics of the specific problem domain. However, whether when dealing with the marginal partitions – Φ and Γ – or the intersect partitions – X – we have a standard set of statistics for each. This is because we have designed the process so that each partition has a common set of columns defined by $A^{(n)}$. As a result, we may use any statistic applied to them for our anomaly definitions.

Perhaps the most common type of anomaly is that defined by the distance of each sample from the mean of its partition. Each partition has, by definition a sample mean, which also happens to be the k-means centroid used when assigning sample membership. And although cluster assignments for Φ and Γ are directly the result of k-means such that no “better” assignment is available, the distance of any given sample from the mean may be quite large as compared to others in the partition. To calculate distance, we may use either Euclidean distance, or when dealing with samples known to be conditioned as ratios, KL-Divergence or Entropy. Then we may easily compute the standard deviation of the collected distances of each sample. Finally, we may then choose some multiple of the standard deviation as a threshold beyond which the sample is classified as an anomaly.

3.8 Visualizing Marginal and Intersect Statistics and Anomalies

In order to review the outcome of the statistical analysis performed by our process, there are several ways to visualize the partitions. Two of particular interest to us are the “classification matrix” and the “strip chart.”

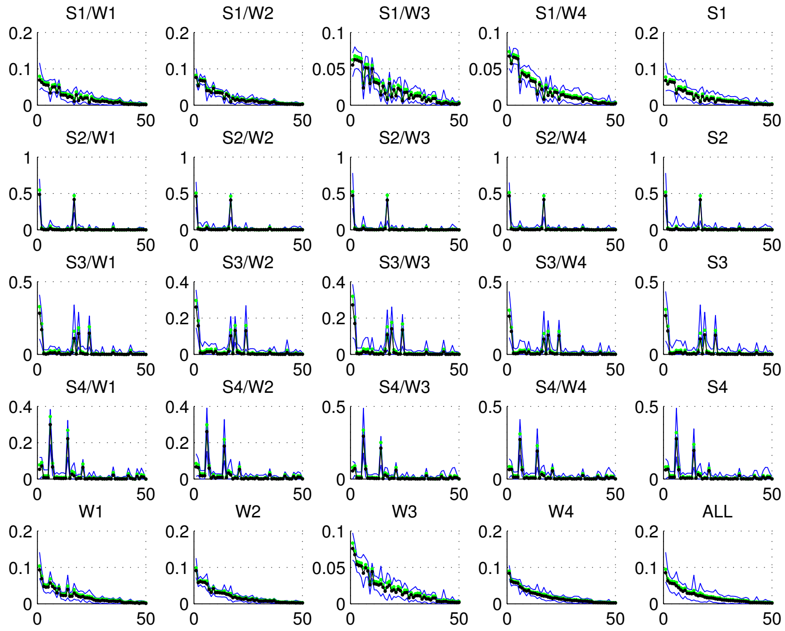


Figure 3.5. Example classification grid showing $||\Gamma|| = 4$ and $||\Phi|| = 4$ where each the resulting $||X|| = 16$ intersect partitions form a 4x4 set of charts with the Γ charts on the right side and the Φ charts on the bottom.

The classification matrix works by allowing us to see some aspect of both the marginal and intersect partitions at the same time. Figure 3.5 illustrates how each of the partitions may be laid out in a visual matrix. Scanning horizontally on each row, we see the X partitions showing variations of Γ for each Φ and with the Γ summary partition on the right. Scanning vertically on each column, we see the X partitions showing variations of Φ for each Γ and with the Φ summary partition on the bottom.

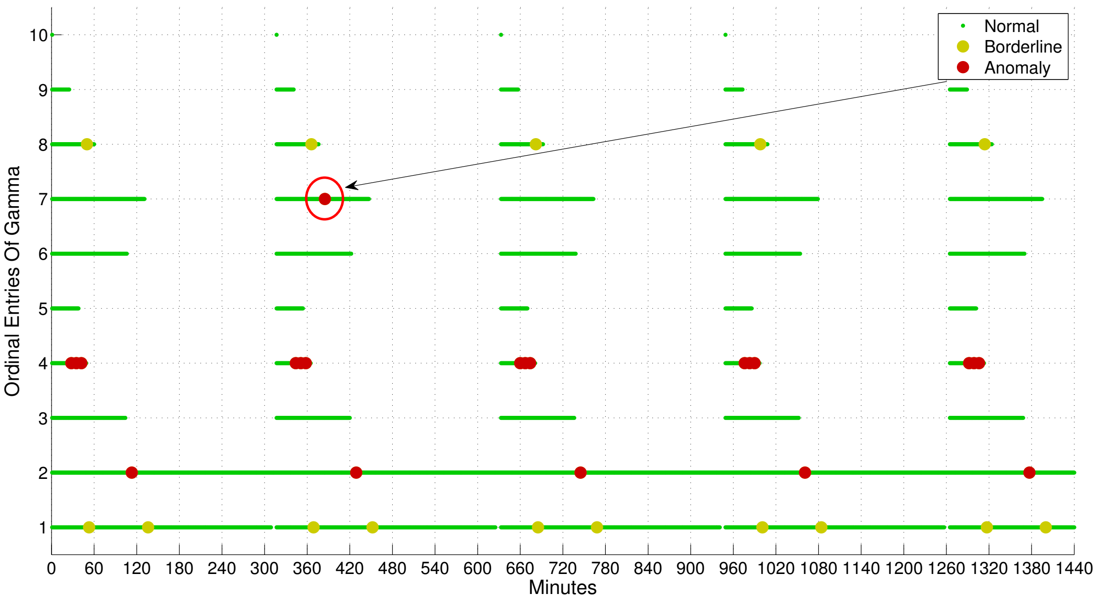


Figure 3.6. Example strip chart showing a row for each Γ and status indicators for each sample placed horizontally according to its occurrence in W.

The strip chart shows some visual indicator of status for each Γ represented by a horizontal chart with the status of each sample in X positioned according to its original location in time corresponding to the time window aggregations of W. The horizontal charts for each Γ align vertically showing how they occur at corresponding times, but in parallel. This visualization is useful for seeing when in time an anomaly occurs as seen in the example of Figure 3.6.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 An Example Of The Process In Action

Consider the scenario where we are an Internet hosting service provider, and are approached by a company that wants to us to host and monitor their new “e-Book Broadcasting” service. They tell us that their service creates a “virtual broadcast channel” for each of the e-Books in their catalog. On each channel they send out the textual content of the assigned e-Book one character at a time in a UDP/IP packet to some unspecified, but distinct destination IP address. Once all of the characters in the e-Book have been sent, they will start to resend the entire book again after some period of delay which they do not specify.

They say that their service is patent-pending, so they do not feel comfortable providing too many details. They will not tell us what books are in their catalog, so we don’t know what languages are used or what genres included, whether fiction or non-fiction, literature or poetry, or any other characteristics of their content. Nevertheless, they want us to monitor the service and let them know if “anything unusual” happens. They further say that they believe the system is working properly, but could really use our help to better understand the its operational dynamics.

We setup their hosting environment and they begin “broadcasting.” Since we can capture the UDP/IP traffic within our network, we begin recording their messages. Based on the little they told us about the service, we apply our process to the message flow.

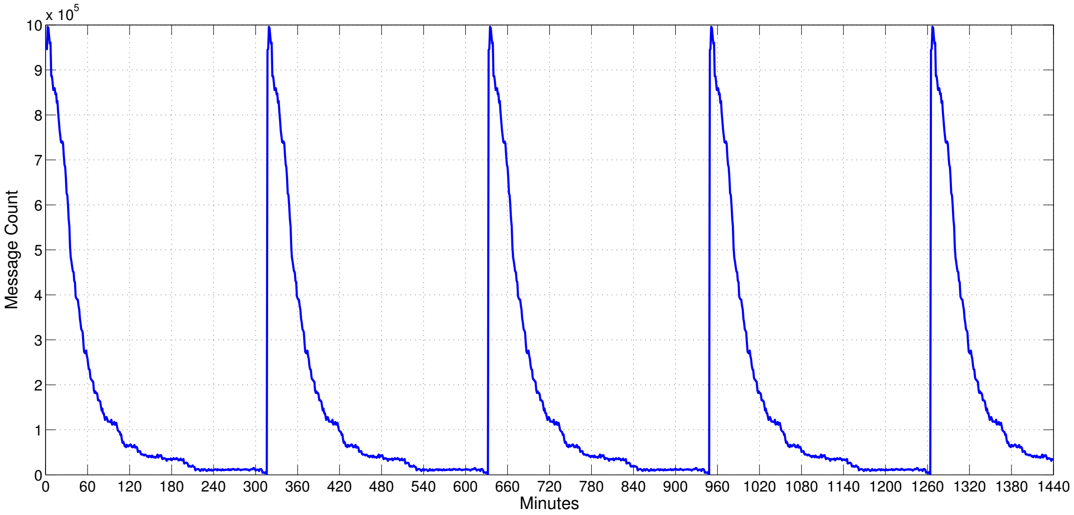


Figure 4.1. The figure displays messages counts on the Y axis captured from the example system for each minute of a 24-hour period on the X axis.

After a full day of operation, we have a captured message flow M as shown in Figure 4.1. The total message flow clearly shows some uniform periodicity, but it is not clear what it signifies. To examine it further, we now need to define the functions $\mathcal{V}_S$, $\mathcal{V}_W$, and $\mathcal{V}_{A'}$ as they apply to our customer's service.

- Function $\mathcal{V}_S$ – We decide to treat each of our customer's logical "channels," one for each e-Book being "broadcast," as our set of sources S. Unbeknown to us, their catalog of e-Books are the collection of public domain documents from our earlier example in Section 2.3.1. As a result, we quickly discover that M has a set of sources where $||S|| = 128$.
- Function $\mathcal{V}_W$ – Our customer is expecting to greatly expand their service, so they have apparently chosen to limit the message rate of each channel. Through simple observation, we can tell that they send about 10K messages per minute for each of their "channels," and we decide to choose a time window size W of 1 minute as well.

- Function $\mathcal{V}_{A'}$ – Knowing that each message simply includes a single character of their e-Book content, we find that they the set of characters in their "alphabet" A consists of 68 symbols – the 26 Roman letters (mono-case), 10 decimal numerals, and 32 punctuation marks. As we saw in the previous analysis of this data set in Section 2.3.1, the distribution of these characters is hyperbolic. Since many of the characters are thus rarely seen, and therefore their statistical contribution minimal, we collapse the least-occurring characters into a single "other" category for the all characters whose combined probability is less than 0.1%. The resulting A' contains only 39 symbols, including "other." Figure 4.2 shows where this threshold falls within the rank-ordered set of symbols by probability.

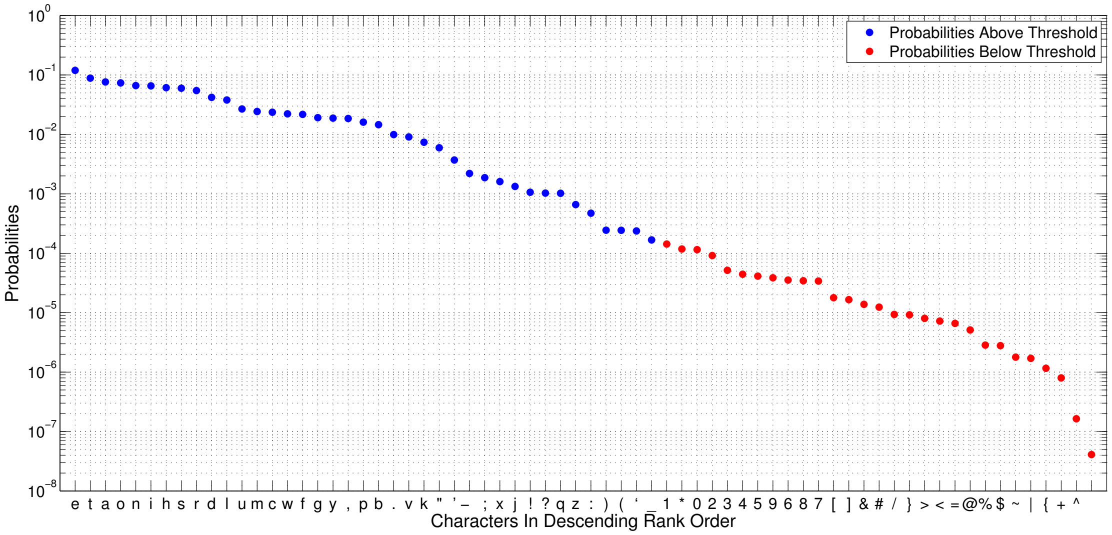


Figure 4.2. This figure shows the characters A occurring in the example data set in descending rank order on the X axis. It shows the probability of each character on the Y axis using the Log scale to help clarify the hyperbolic character of the distribution. The change in color of the data points into two groups shows the point at which the remaining probability of the least-frequent characters totals less than 0.1%.

With S, W, and A', we are able to define our marginal summaries of M by volume and ratio as defined by the process: V_S, V_W, R_S, and R_W. By applying both PCA and then k-Means to each, we settle for simplicity on a value of $k = 10$ to define our source types Γ_V and Γ_R, as well as our window types Φ_V and Φ_R.

Table 4.1. Count Of Documents For Each Cluster In Γ

Cluster	Count
γ_1	28
γ_2	20
γ_3	18
γ_4	18
γ_5	12
γ_6	10
γ_7	9
γ_8	7
γ_9	5
γ_10	1

Looking closely at Γ_R, we note that one of the clusters contains only one (1) source γ_{10} as shown in Table 4.1. The significance of this is that a single document stands out from all others in the *eigenspace* characterized by the probability distribution of A'. A smaller value of k would have clustered this document along with others, but it would likely have been far from the cluster centroid and been characterized as a "source anomaly."

We return to the customer with this document as "something unusual." After reviewing the contents of the document in question, they tell us that they found the content of the document to have been corrupted. Instead of an actual e-Book, the document contained the HTML content of an error page they had received from their content provider when they were originally building their catalog. They thanked us for helping them find this error, noting that the volume of content in their catalog made it impossible for them to review and verify all documents.

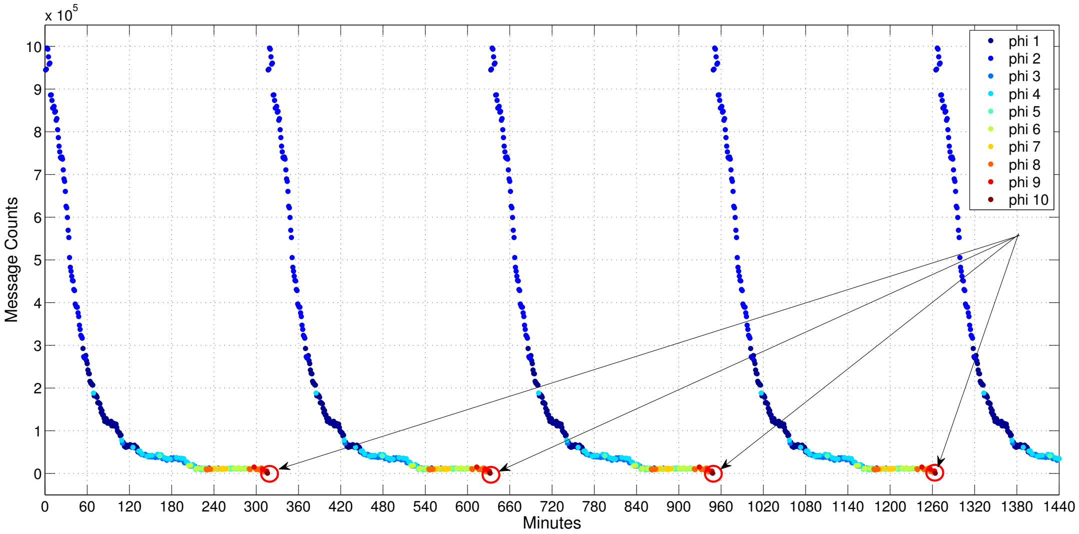


Figure 4.3. This figure uses the categorizations of Φ to color-code the 1-minute time periods of M on the X axis and the total message counts seen in each period on the Y axis.

Next we look at Φ_R and note similar situation where a cluster has only four (4) periods during the course of the day. Color coding the time periods to our original Figure 4.1 by Φ and circling the four (4) time periods in the smallest cluster in Figure 4.3, we see that this cluster ϕ_{10} also corresponds to the minimum volume window periods.

We return to the customer noting these minimal message rate windows ϕ_{10}. They reveal that they wait for all the characters for all "broadcast channels" to be sent before restarting each channel. The customer noted that the uniqueness of the ϕ_{10} cluster suggested that a minimal statistical sample that had such little data as to poorly represent the typical distribution of the single, longest document in their catalog. Again, they thanked us for the insight into their system, recognizing that their approach overall poorly utilizes their available bandwidth.

As with γ_{10}, the nominal elements of the ϕ_{10} cluster are such that a smaller value for k would likely have resulted in these time periods being categorized as "window anomalies."

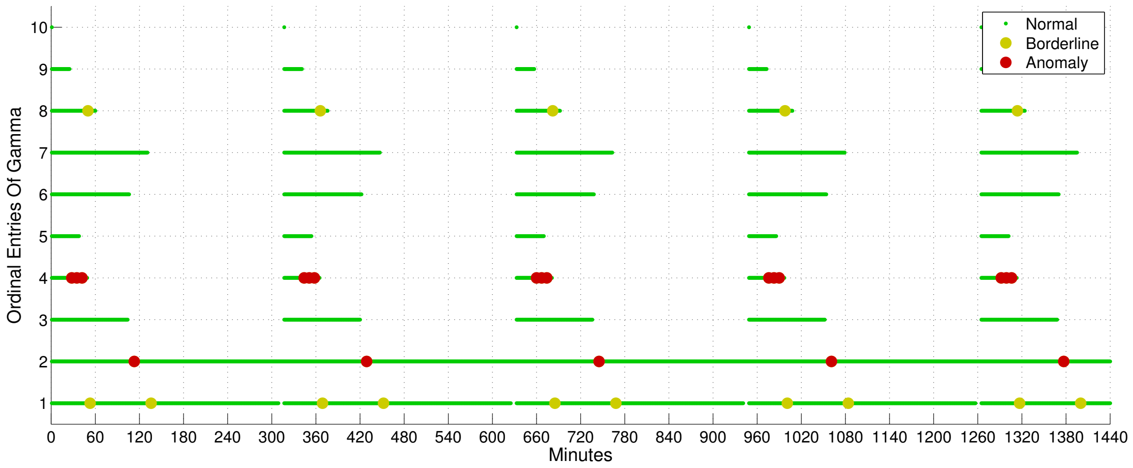


Figure 4.4. This figure uses the categorizations of Φ to color-code the 1-minute time periods of M on the X axis and the total message counts seen in each period on the Y axis for day 1.

Taking both Γ_R and Φ_R together to find intersects X_R, we consider the KLD of each intersection sample matching the elements of the intersects X compared to average. Figure 4.4 shows that a small number of these intersections (in red) fall outside of a threshold of 2 or about half of the entropy value H of the information measure for A' of 4. The "red" items in Figure 4.4 correspond to "intersect anomalies" that we bring to the customer's attention.

While waiting for the customer to investigate the cause of the "intersect anomalies" for source types γ_2 and γ_4, we continue capturing the message flow for the service

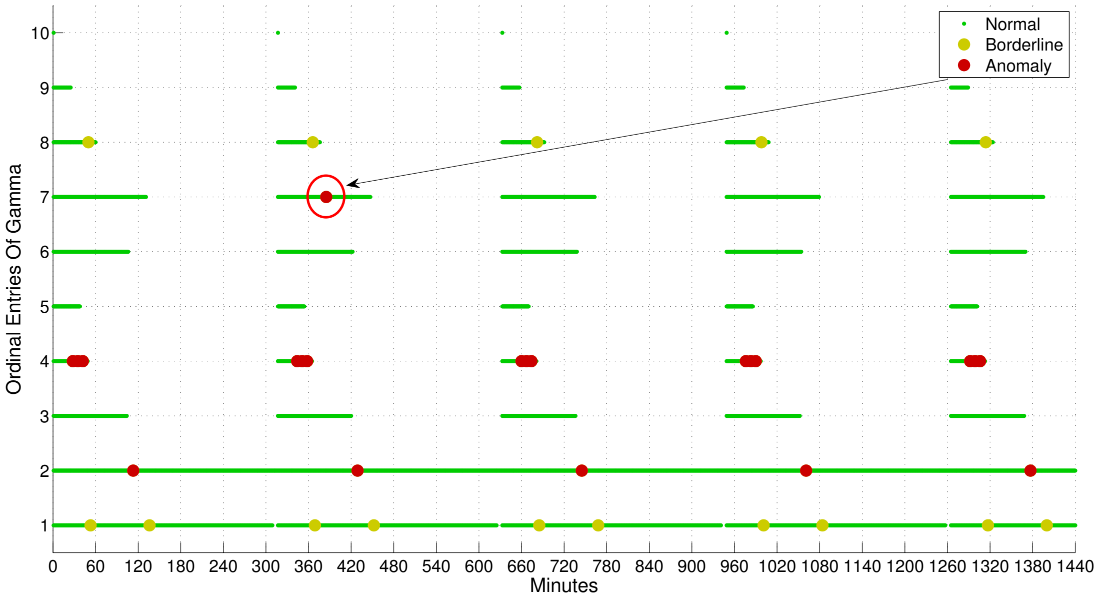


Figure 4.5. This figure uses the categorizations of Φ to color-code the 1-minute time periods of M on the X axis and the total message counts seen in each period on the Y axis for day 2, with special note of a new "intersect anomaly" as compared to day 1.

for a second day. In doing so, we notice that a new "intersect anomaly" occurs as noted in Figure 4.5.

After bringing the new anomaly detected on day 2 to the customer's attention for γ_7, they investigated their logs and find that during the second broadcast of one of the documents in the group categorized by γ_7 failed was, in fact, *not* "broadcast" as expected. The customer, again, thanked us for noting the "something unusual" in their system's behavior.

4.2 Study: Cellular Wireless Signaling

"Signaling" is a standard part of connection-oriented communications protocols and is generally transparent to the application-level visible to operators of M2M applications within the IoT, being several levels removed in a tiered services environment with multiple layers of abstraction hiding its presence. For older cellular wireless networks such as "Global System for Mobile Communications" (GSM) [63], this signaling layer is provided by "Signaling System 7" (SS7) [64]. In the newer "Long-Term Evolution" (LTE) network standard [65], the signaling layer is provided by the Diameter protocol [66].

With SS7, a wireless device communicates with a local cell tower for the purpose of registering its presence, establishing a connection through which to send its application data, and other activities. The tower follows the signaling protocol to communicate with other network elements on behalf of the device to authenticate it, authorize its actions, and perform those authorized actions. In the case of SS7, the protocol defines over one hundred operations – think predicates – invoked on behalf of a device – think subject – by one network element and responded to. Device subjects with invocation/response predicates form the basis for a linguistic dialog that may be studied to discern patterns of behavior for the devices that give rise to the signaling activity among the network elements of the communications infrastructure.

Using methods developed in [67] and [68], we are able to capture the raw network packets exchanged among cellular network elements and convert them into a stream of timestamped subject/predicate pairs that are the essential linguistic components of the SS7 protocol.

For this investigation, we consider several subsets of the larger population of devices whose communications are captured and made available using [67] and [68]. In particular, we consider the SS7 traffic for the devices of several commercial M2M applications over a one week period as seen in Table 4.2. The selected applications serve very different business purposes, vary in population of two orders of magnitude, and generated a number of messages that range over three orders of magnitude.

Table 4.2. M2M Applications Considered

Type	Devices (D)	Messages (M)	Ratio (M/D)
Security	374,268	167,759,619	448.2
Real Estate	291,654	80,044,881	274.5
Vehicle Tracking	8,972	3,091,959	344.6
Oil & Gas	1,159	903,341	779.4

4.2.1 Considering Aggregate Messages Over Time

The devices in our selected applications communicate over the course of the target week to to produce the message counts shown in Table 4.2. So we next consider the message arrival over time to better understand their behavior.

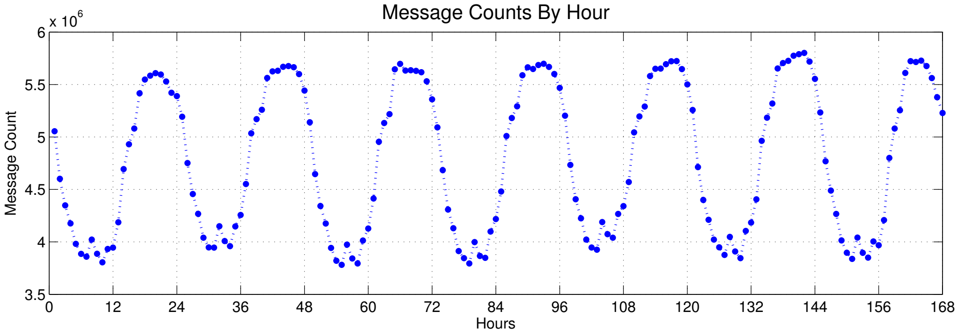


Figure 4.6. The chart shows the total SS7 message count on Y for every hour of the target week on X for all devices in the data set.

The target M2M application devices' SS7 traffic represent only a portion of that for over 1.6 million wireless devices sharing the same network resources. The combined traffic seen using [67] for the target week is shown if Figure 4.6. In it, we easily observe the diurnal character of the SS7 traffic as more activity by the device population takes place during the day and less at night.

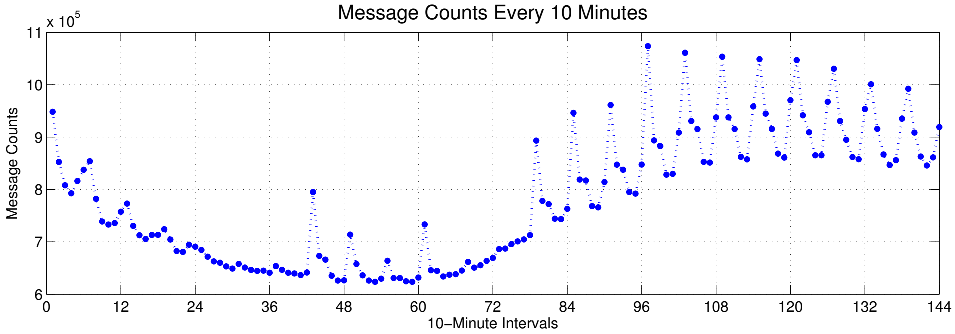


Figure 4.7. The chart shows the total SS7 message count on Y for each 10-minute interval of a single day on X for all devices in the data set.

Looking more closely at a single day with a smaller time increment of 10-minutes, we further observe additional variations such as hourly spikes in the message counts as seen in Figure 4.7.

As stated, Figures 4.6 and 4.7 represent message frequency counts for all application traffic captured, but as might be expected, not all applications exhibit have the same reporting profile over time nor the same distribution of message types. Figures 4.8a-d and 4.9a-d show the same weekly and single day message frequency counts for each of the target M2M applications in this study. These figures break out the data for each application into separate charts to allow the large differences in Y-values to vary independently.

In Figures 4.8a-d, we observe that the all four target applications exhibit some nominal diurnal properties, but have very different amplitudes. In the case of the “Security” and “Oil & Gas” applications, we saw in Table 4.2 that they had the highest message count per device (M / D) values. We also see in corresponding Figures 4.8a and 4.8d that these applications also have much higher minimum message rates relative to their maximum rates that help explain the higher messages count per device. By contrast, the “Real Estate” and ”Vehicle Tracking” applications in

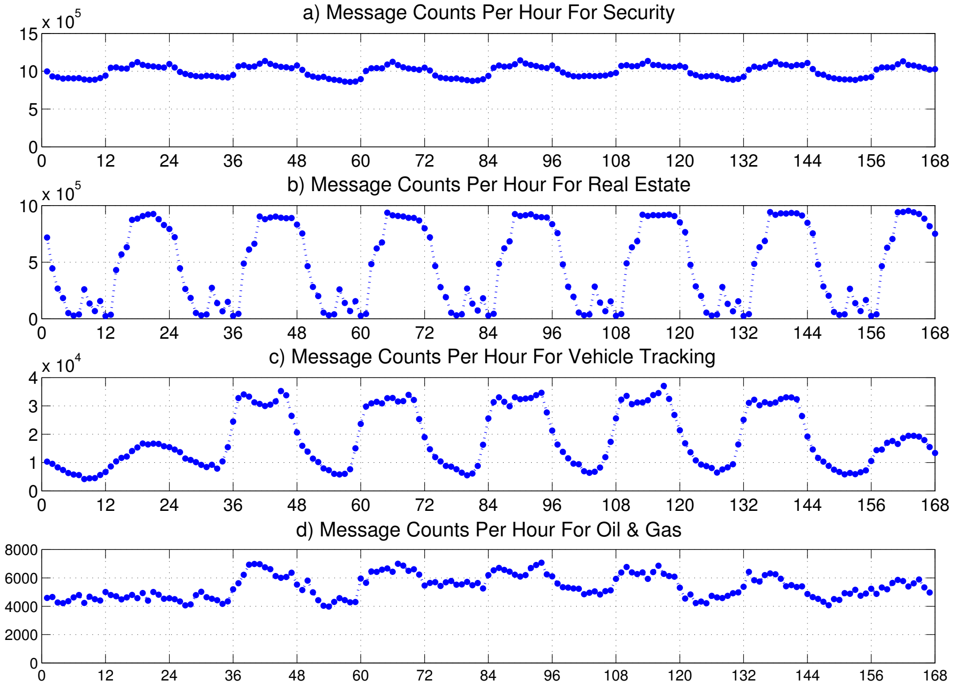


Figure 4.8. Charts “a” through ”d” show the total SS7 message count on Y for every hour of the target week on X for each of the four labeled target M2M applications.

Figures 4.8b and 4.8c experience minimum message rates approaching zero which helps lower the overall message count per device for each.

As the time interval decreases in size, we observe that some applications exhibit more “bursty” variation as shown in Figures 4.9b and 4.9d as compared to Figures 4.9a and 4.9c. In the case of the application if Figure 4.9b (i.e., “Real Estate”), the pronounced message-count spikes correspond to those in the message counts for all applications from Figure 4.7.

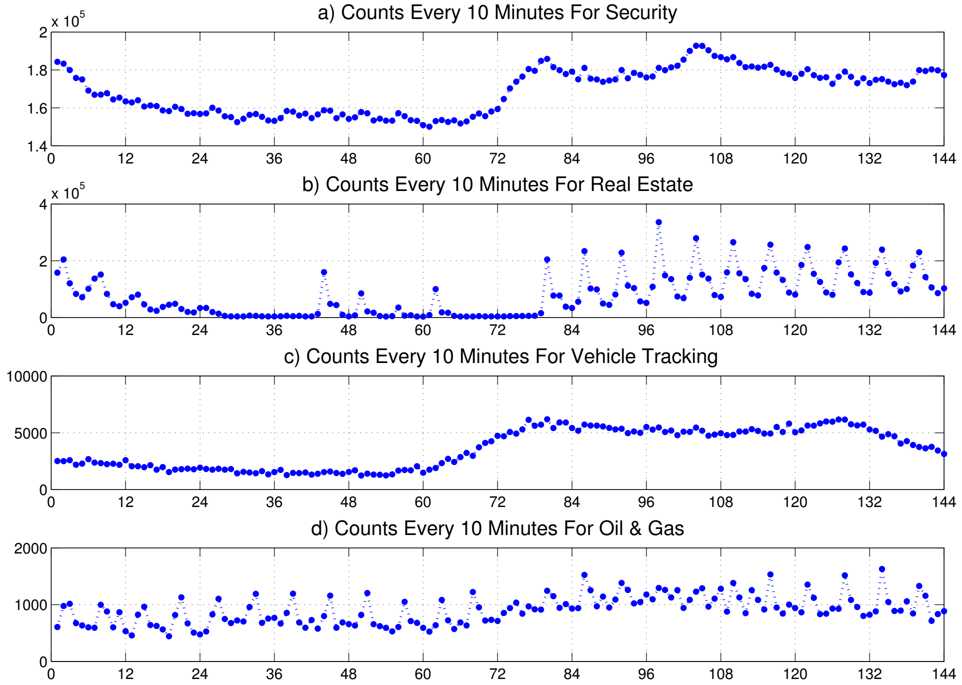


Figure 4.9. Charts “a” through ”d” show the total SS7 message count on Y for each 10-minute interval for a selected day on X for each of the four labeled target M2M applications.

4.2.2 Considering Message Types

Of course, all messages arriving over time from the applications are not the same type. Understanding what type of messages arrive and their frequency is the next level of analysis we need to perform.

For our purposes, we construct an *alphabet* for SS7 signaling based on the network elements initiating and receiving messages, the operation code of the message if an invocation or response, as well as an error code when the response is unsuccessful. The official SS7 specification identifies 106 unique operation codes and 60 error codes

[69]. However, in our sample week, we observe only 19 operation codes and 10 error codes, resulting in an *alphabet* of 72 symbolic message types.

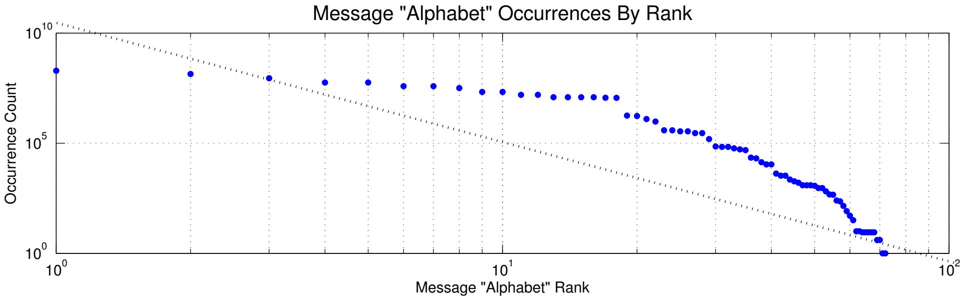


Figure 4.10. The chart shows the number of SS7 message-type occurrences on Y for each message type in rank order on X.

Figure 4.10 shows evidence of the hyperbolic distribution one should expect from the frequency distribution of language symbols by plotting the message counts for each of the symbols in rank order on the log/log scale. The result is an approximately-linear arrangement of the data. It is perhaps worth noting that the qualification of *approximately* linear results from the fact that the data set, as well as the symbols set, is relatively small, and as such, these results are similar to those found in other language alphabets. While it is considered an empirical fact that language symbol frequencies follow such hyperbolic distributions, this behavior is observed most clearly at the limit [41].

Looking at the probability distribution of message types for each application in Figure 4.11, we easily see differences in how the devices communicate with respect to the average of all 1.6 million devices and with respect to each other. By using probability in this case, instead of total message count, we are able to compare how each application communicates regardless of the size of the device population in each.

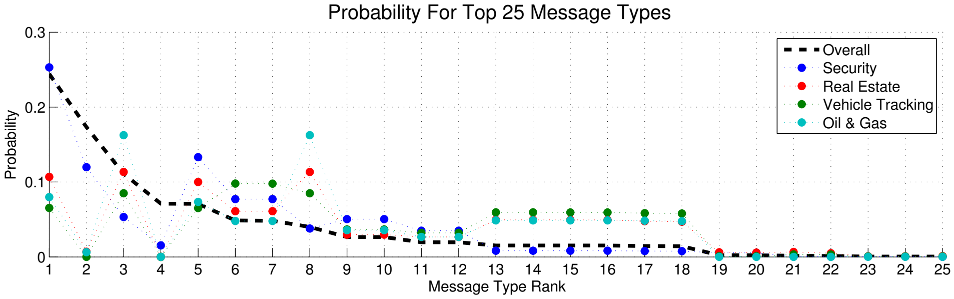


Figure 4.11. The chart shows the number SS7 message type probabilities on Y for each of the top 25 message types by rank on X.

At this stage, we do not care *what* the message types are but only their rank as a function of message type frequency from Equation 2.11 so that we can compare how the use of the message types differs by application. Regardless of *what* the message types represent operationally, it is clear that some the applications have similarities as well as differences from each other in terms of their use of these message types.

4.2.3 Considering n-Grams Based On Message Type

Without elaborating on the specifics of the actual SS7 message types, we can state that the order of occurrence of different messages for a specific device represent different operational scenarios for the application. Similarly, the sequence of message types generated by devices within an application may differ as well. Furthermore, two applications that have similar message type frequencies may have different results from each other when considering message-type sequence frequencies. Such message-type sequences can be considered using an n-gram analysis.

For the target week, we stated that a total of 72 message types were observed. Over the course of a year, the source environment has encountered very few additional message types, so the 72 types here are very representative the operational characteristics of a far longer time frame.

Table 4.3. Possible vs. Actual n-Gram Occurrences for SS7

N	$\|\|A\|\|^N$	=	Possible (P)	Actual (A)	Percent (A/P)
1	72^1	=	72	72	100.00%
2	72^2	=	5,184	1,080	20.83%
3	72^3	=	373,248	4,516	1.21%
4	72^4	=	26,873,856	12,867	0.05%

With 72 message types, we have 72 possible "1-grams." By passing through the data set for each unique device, we can identify successive pairs of messages whose occurrence represent possible "2-grams." It is possible to the record pair of types as a 2-gram either as a combination or permutation – in our case, we use permutations.

By expanding the size of the sequence from 2 to 3 and 4, we likewise identify type occurrences that we can call "3-grams" and "4-grams." Counting unique permutations of 2-, 3-, and 4-grams, we arrive at the data found in Table 4.3. In this table we can see that the *possible* number of observed n-grams at each value of n grows exponentially, but the *actual* number of permutations that occur is dramatically smaller.

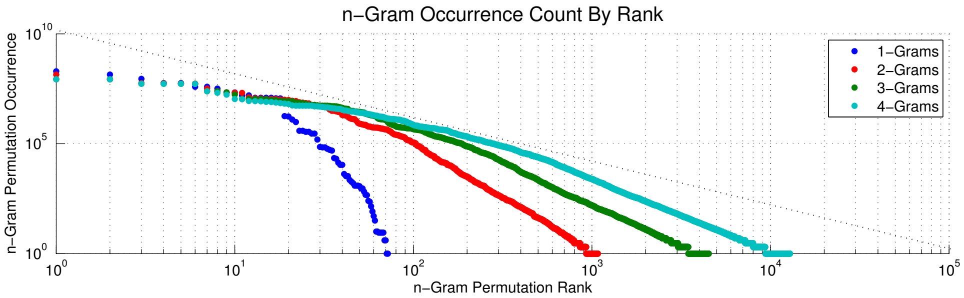


Figure 4.12. The chart shows the number of occurrences on Y for each n-gram permutation in rank order on X for 1-, 2-, 3-, and 4-grams.

As n grows, the frequency of n-gram permutations more closely matches the expected linear relationship on a log/log scale as expected of a hyperbolic distribution as shown on Figure 4.12. Note that the 1-gram data on the chart is duplicated from Figure 4.10 for reference and to show how 2-, 3-, and 4-grams progressively "straighten" their curvature.

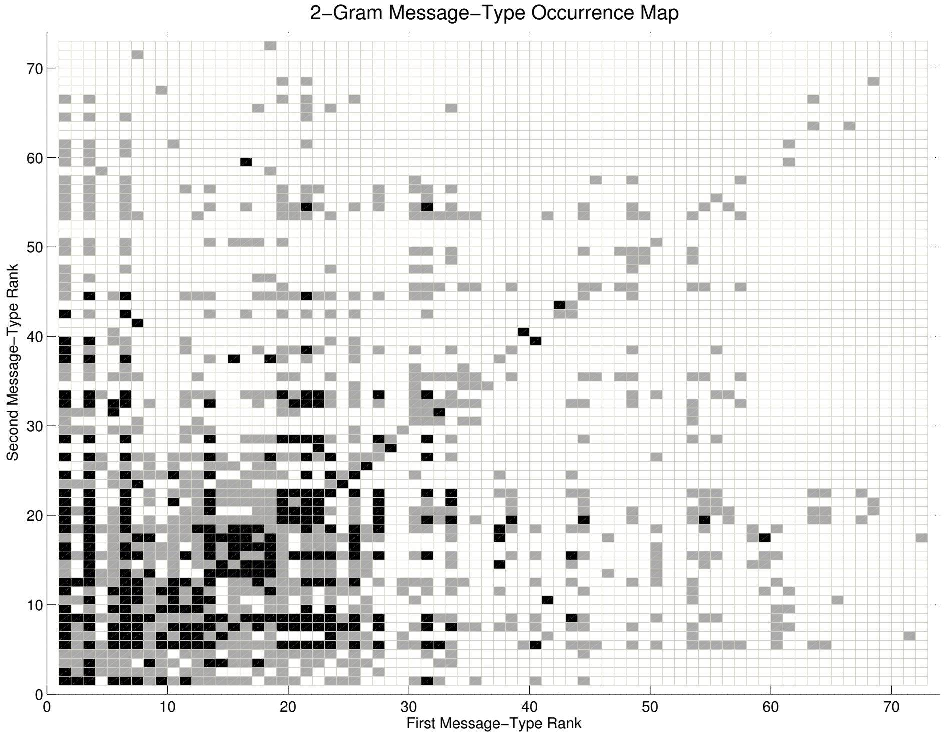


Figure 4.13. The heat map indicates the occurrence of 2-grams defined by a starting message type on X and an ending message type on Y where the black squares represent 2-gram occurrences for the "Vehicle Tracking" application, gray squares represent occurrences for all applications, and the white squares represent no occurrence; note that message types are represented by their rank index from 1 to 72.

Figure 4.13 visualizes the sparsity of message-type permutations that occur in practice by showing which starting/ending message-type permutations occur as 2-

grams within the overall data set. In this figure, the white (blank) cells indicate permutations that did not occur in the data set, while the gray cells indicate those that did occur. In addition, the black cells are the subset of all 2-grams that occurred for the "Vehicle Tracking" application. Notice that the the permutations are not symmetrical; that is, a given pair of message types (e.g., 'A' followed by 'B') may occur but not the reverse (e.g., 'B' followed by 'A'), either for a specific application, like "Vehicle Tracking" in black, or all applications combined in gray.

4.2.4 Mapping n-Grams To Operational Semantics

Until now we have been focusing on the probabilistic differences of specific M2M applications and the overall system within which they operate. We have been considering n-gram frequencies with simple rank assignments to each occurrence, regardless of their meaning. We are now ready to reveal the semantics of the n-grams and consider what they say about the behavior of the M2M devices that give rise to them within the data set.

Before we can do this, we must first briefly describe the network elements involved in SS7 signaling. All devices communicating over the network must exist in the inventory database of a "Home Location Register" (HLR). When a device begins communicating with a cell tower, it must first authenticate itself. The cell tower forwards the authentication request, called an "invocation," to a "Visitor Location Register" (VLR) which handles all such traffic for a group of towers and which keeps an inventory of all devices currently "visiting" a tower within its domain. The VLR submits a "Send Authentication Info" (code 56) invocation to the HLR to which the device belongs. If the device is authorized to communicate over the network, the HLR responds in the affirmative, otherwise it might respond an "Unknown Subscriber" (code 01) error code. If the VLR does not current have the device in its register, it will next submit an "Update Location" (code 02) invocation to the HLR. In response, the HLR will contact the previous VLR by sending a "Cancel Location" (code 03) invocation, allowing the VLR to remove the device from its inventory, and will send an "Insert Subscriber Data" (code 07) invocation to the new VLR. Table 4.4 summarizes the most common events in the data set that we will be considering.

Table 4.4. Common SS7 Event Codes

Code	Description
02	Update Location
03	Cancel Location
07	Insert Subscriber Data
23	Update GPRS Location
56	Send Authentication Info

With the advent of cellular data communications using the GPRS protocol at a level above the SS7 signaling layer, instead of enhancing the function of the VLR, a new type of network element was introduced side-by-side with the VLR called the "Serving GPRS Support Node" (SGSN). Devices that communicate using (typically) IP communication must also be authenticated by the SGSN which generate a similar sequences of "Send Authentication Info," "Update Location," "Insert Subscriber Data," and "Unknown Subscriber" invocations, responses, and errors.

In the rest of this paper, we will use a short-hand symbolic reference for SS7 events following the pattern "<event-type><origin-code><destination-code><event-code>" where event types are "I" for invocations, "R" for responses, "E" for errors, and where origin and destination codes are the first letter of the network element type (i.e., "V" for VLR, "H" for HLR, and "S" for SGSN). Thus a VLR sending a

"Send Authentication Info" invocation to an HLR takes the form IVH56, the HLR's successful response is RVH56, and an "Unknown Subscriber" error is EVH01.

We saw previously the probability distributions of the 1-grams for all applications together and our four selected applications in Figure 4.11, focusing only on the top 25 and likewise most-significant values. As highlighted in Table 4.5, the single most frequent 1-gram for all applications, and a relatively high-probability one for our selected applications, is the "Send Authentication Info" invocation by a VLR (IVH56) with the second most frequent 1-gram for all applications being the "Unknown Subscriber" error (EVH01). The third and fourth 1-grams by rank are the "Send Authentication Info" invocation by SGSN elements (ISH56) followed by the "Unknown Subscriber" error as well (ESH01).

Table 4.5. Top 10 1-Gram Symbols and Probabilities

Rank	Symbol	Probability
1	IVH56	0.2443
2	EVH01	0.1734
3	ISH56	0.1111
4	ESH01	0.0710
5	RVH56	0.0710
6	IVH02	0.0486
7	RVH02	0.0482
8	RSH56	0.0400
9	IVH07	0.0266
10	RVH07	0.0266

The presence of "Unknown Subscriber" errors as the third and fourth highest ranking 1-grams is surprising and we will investigate further. However, it is clear from Figure 4.11 that the four selected M2M applications have much lower, almost zero, probability of these errors, with the exception of "Unknown Subscriber" errors in response to VLR invocations (`EVH01`) only for the "Security" application.

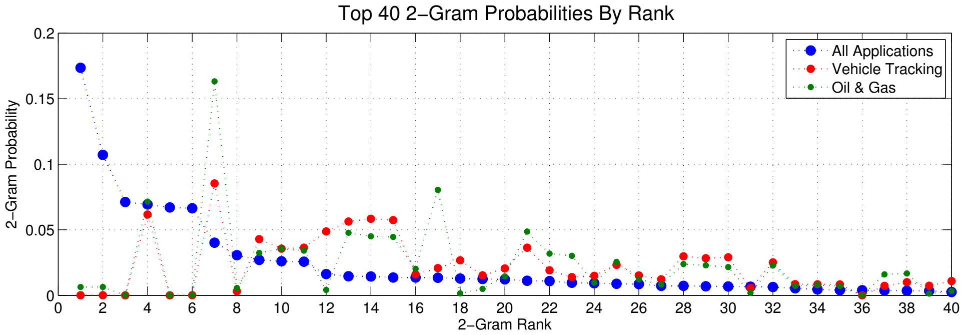


Figure 4.14. The chart shows the probability of occurrence on Y for the top 40 2-gram permutations in rank order on X for all applications combined, as well as for the "Vehicle Tracking" and "Oil & Gas" applications.

Figure 4.14 shows visually and Table 4.6 empirically that the first and second ranked 2-grams are the `IVH56-EVH01` pair followed by its reverse permutation of `EVH01-IVH56`, both related to VLR activity. The third ranked 2-gram is the `ISH56-ESH01` pair, but its reverse permutation drops down to rank 36 – not shown in the table. Instead, the most-likely event paired with `ESH01` is `IVH56` at rank 5. The reason for this will become more apparent when considering 4-grams.

The "apriori principle" used in association rule mining [70] says effectively that a high-probability combination of length $n + 1$ must be constructed using high-probability combinations of length n. So it stands to reason that the highest-probability 4-grams would consist of high-probability 2-grams as well. And by looking at the top 4-grams, we see a more clear picture of frequent communication patterns occurring within the data set.

Table 4.6. Top 20 2-Gram Symbols and Probabilities

Rank	Symbol Sequence	Probability
1	IVH56-EVH01	0.1736
2	EVH01-IVH56	0.1071
3	ISH56-ESH01	0.0712
4	IVH56-RVH56	0.0694
5	ESH01-IVH56	0.0670
6	EVH01-ISH56	0.0664
7	ISH56-RSH56	0.0401
8	RVH56-IVH56	0.0306
9	RVH56-IVH02	0.0270
10	RVH07-RVH02	0.0260
11	IVH02-IVH07	0.0256
12	IVH02-RVH02	0.0162
13	RSH07-RSH23	0.0145
14	ISH23-ISH07	0.0144
15	ISH07-IHS03	0.0136
16	IVH07-RVH07	0.0135
17	RSH56-ISH56	0.0134
18	RVH02-IVH02	0.0127
19	RVH02-IVH56	0.0125
20	IVH07-IHV03	0.0121

As we saw with the figure and table for 2-grams, Figure 4.15 shows visually and Table 4.7 empirically the highest-ranking 4-gram probabilities. The first two ranks correspond almost equal probabilities for the two possible permutations of alternating IVH56 and EVH01 symbols. The third through sixth ranks correspond to the almost equal probabilities for each phase of the four-symbol sequence of IVH56-EVH01-ISH56-ESH01.

The first two ranks represent the common occurrence within the data set of devices whose only interaction within the SS7 layer is the repeated request to authenticate via a VLR (IVH56) and the denial of that request (EVH01). The next four

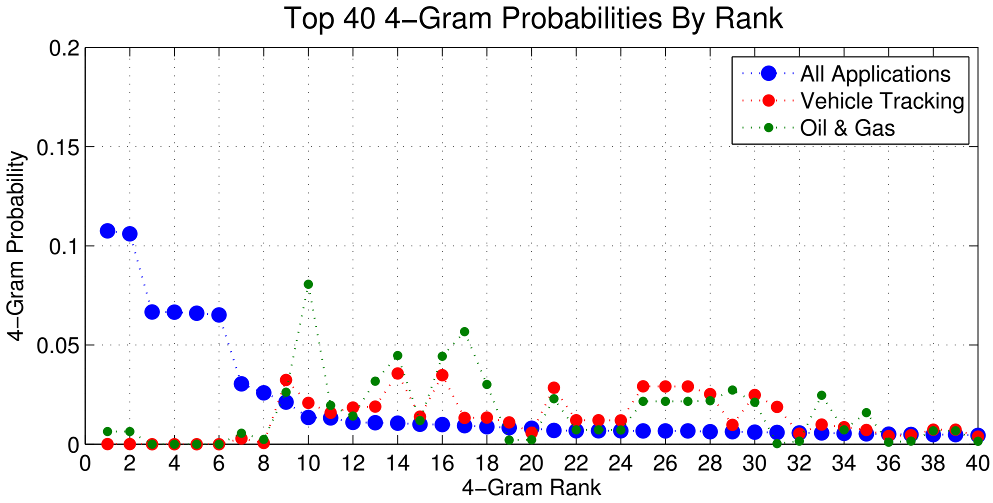


Figure 4.15. The chart shows the probability of occurrence on Y for the top 40 4-gram permutations in rank order on X for all applications combined, as well as for the "Vehicle Tracking" and "Oil & Gas" applications.

ranks represent a similar scenario where devices repeat the same request for both a VLR and its paired SGSN. These two scenarios relate to 1) devices whose use of the network may only include SMS communication and thus do not engage with an SGSN as part of their operation, and 2) devices who use GPRS communication and therefore must authenticate both via both a VLR for basic network access and SGSN for data-communications access.

The significance of these scenarios is that the devices in question are receiving "Unknown Subscriber" responses because their service has been terminated by the carrier. Nevertheless, the devices continue to try and connect to the network and do so in some cases at a higher frequency than they would had their authentication request been accepted. Were the devices in question to be consumer cell phones as might have been assumed by the designers of SS7 and GPRS networking protocols, the termination of service would typically be associated with the consumer's discontinuing to use the phone. Even if the phone remained powered on, its battery would eventually

Table 4.7. Top 20 4-Gram Symbols and Probabilities

Rank	Symbol Sequence	Probability
1	IVH56-EVH01-IVH56-EVH01	0.1075
2	EVH01-IVH56-EVH01-IVH56	0.1060
3	IVH56-EVH01-ISH56-ESH01	0.0666
4	ISH56-ESH01-IVH56-EVH01	0.0666
5	EVH01-ISH56-ESH01-IVH56	0.0660
6	ESH01-IVH56-EVH01-ISH56	0.0652
7	IVH56-RVH56-IVH56-RVH56	0.0304
8	RVH56-IVH56-RVH56-IVH56	0.0259
9	IVH56-RVH56-IVH02-IVH07	0.0212
10	ISH56-RSH56-ISH56-RSH56	0.0134
11	IVH02-IVH07-RVH07-RVH02	0.0132
12	RVH56-IVH02-IVH07-IHV03	0.0110
13	ISH56-RSH56-IVH56-RVH56	0.0108
14	ISH56-RSH56-ISH23-ISH07	0.0106
15	RVH56-IVH02-IVH07-RVH07	0.0101
16	RSH56-ISH23-ISH07-IHS03	0.0098
17	RSH56-ISH56-RSH56-ISH56	0.0093
18	IVH56-RVH56-ISH56-RSH56	0.0088
19	RVH02-IVH56-RVH56-IVH02	0.0083
20	RVH07-RVH02-IVH56-RVH56	0.0080

die and it would stop attempting to connect to the network. However, it is quite common for M2M devices to be connected to a permanent power source and to be embedded in some larger piece of equipment (e.g., wired into a vehicle), perhaps even unknown to or forgotten by the equipment operator. As a result, the termination of service at the carrier level would be insufficient to ensure that M2M device stops operating. In fact, such an operational condition may not even have been considered by the M2M device designer.

Moving beyond the "Unknown Subscriber" scenarios, the n-gram analysis gives us additional insight into the operational dynamics of our selected M2M applications.

Since these applications have few, if any, devices with discontinued service, their high probability n-grams involve many other activities.

To illustrate, we look closely at the "Vehicle Tracking" and "Oil & Gas" applications whose 1-, 2-, and 4-gram probabilities are seen in Figures 4.11, 4.14, and 4.15. The 1-gram probabilities for these applications appear similar except for ranks 3, 6, 7, and 8, corresponding to events `ISH56`, `IVH02`, `RVH02`, `RSH56`, respectively. It is also evident that, despite a general decrease in probability for these events overall, the probability of the invocation of each type of event closely matches the successful response for each application.

The 2-gram probabilities likewise are similar for these two applications as shown in Figure 4.14 with significant differences seen at ranks 7, 12, and 17, corresponding to event pairs `ISH56-RSH56`, `IVH02-RVH02`, and `RSH56-ISH56`, respectively. Note that these 2-grams contain the same events primitive events that were different between the applications at the 1-gram level. What we see at the 7th rank are almost twice as many "Send Authentication Info" invocation/response pairs by SGSNs on behalf of the "Oil & Gas" devices than for the "Vehicle Tracking" devices even though the 1-gram distribution shows that only roughly 30% more invocations are submitted. The 12th ranked 2-gram indicates that the "Vehicle Tracking" application has a large number of successful "Update Location" invocation/response pairs compared a negligible number for the "Oil & Gas" application. Furthermore, the 17th rank 2-gram shows that one successful "Send Authentication Info" invocation/response pair is followed by another almost 50% of the time for the "Oil & Gas" application as compared to perhaps 20% of the time for the "Vehicle Tracking" application.

Finally, the 4-gram probabilities in Figure 4.15 bring into sharper focus how these event sequences differ between the applications. Here we see that the most significant difference between the two applications is the dramatically higher probability for the “Oil & Gas” application at ranks 10 and 17, corresponding to symbol sequences `ISH56-RSH56-ISH56-RSH56` and `RSH56-ISH56-RSH56-ISH56`, respectively, which represent the two phases of alternating symbols `ISH56` and `RSH56`.

Taking the 1-, 2-, and 4-gram observations together, a picture emerges of behavioral difference between the “Vehicle Tracking” and “Oil & Gas” applications.

As the name suggests, the “Vehicle Tracking” application employs devices that are attached to vehicles which, when moving from place to place, require registration with different towers, and therefore result not only in a high occurrence of “Send Authentication Info” messages sent on behalf of the devices by VLRs, but also the need for “Update Location” messages that are triggered when the current VLR of record changes. That the ”Update GPRS Location” message frequency by SGSNs is not high in the list of frequent events suggests that the devices may be moving more frequently than their need to employ the services of an SGSN for data communications. And in fact we know that this particular application has a device configuration designed for very low-frequency reporting. Nevertheless, certain signaling activity must take place “just in case” the device decides to send a payload.

In the case of the “Oil & Gas” application, the M2M devices employed are in fixed geographic locations and use cellular wireless communications to overcome the limited availability of terrestrial options at remote service sites. The high frequency of repeated alternating `ISH56` and `RSH56` events reflect the fact that the devices frequently connect and report their application payload with very little additional interaction with the network required. That these devices still need to authenticate and update the network elements with their presence corresponds to the need for the network refresh their their inventory of “visiting” devices, given the design expectation that mobile devices might leave the network without notice.

4.2.5 Considering n-Grams for Time-Bounded "Motifs"

When performing n-gram analysis on a corpus of textual documents, it makes sense that there is a first and last n-gram instance in each document consisting of the first and last n characters or words, and there are *not* n-gram instances for characters or words that *span* documents. However, when working with messages sequences of the type generated by M2M devices, notions like document boundaries are not clearly defined.

Casual inspection of the data flow such as that in our M2M signaling data reveals that messages often arrive in bursts with over several seconds separated by long intervals measured in minutes or hours.

So far, our n-gram occurrences have been collected regardless of the inter-message interval, but like the last characters/words in one text document and the first characters/words in a following text document, the last messages in one burst and the first messages in a following burst may have very little informational meaning, and counting them in our statistics could skew the results.

For this reason, we consider the bursts as sequences messages with some minimum interval as "motifs" and may perform our n-gram analysis on those messages which occur only within each motif in the data set, discarding message sequences across motifs.

By imposing this construct on our data, we immediately see a change in the n-gram occurrences. Table 4.8 shows an example of the number of unique n-grams occurring with the "Vehicle Tracking" application are significantly reduced when considering only motifs separated by 60 seconds or more.

Table 4.8. All vs. "Motif" n-Gram Counts for "Vehicle Tracking"

N	All (A)	Motif (M)	Percent (M/A)
2	302	262	86.8%
3	862	695	80.6%
4	2,095	1,460	69.7%

4.2.6 Considering k-Means Clustering Scenarios

As outlined in Chapter 3, our process involves the composition of time-series data set along several dimensions in order to characterize its behavior. We have already examined the linguistic properties of M2M applications using signaling data using n-gram and "motif" analysis, so we are ready to discover different ways to group similar devices and time intervals together using these characteristics.

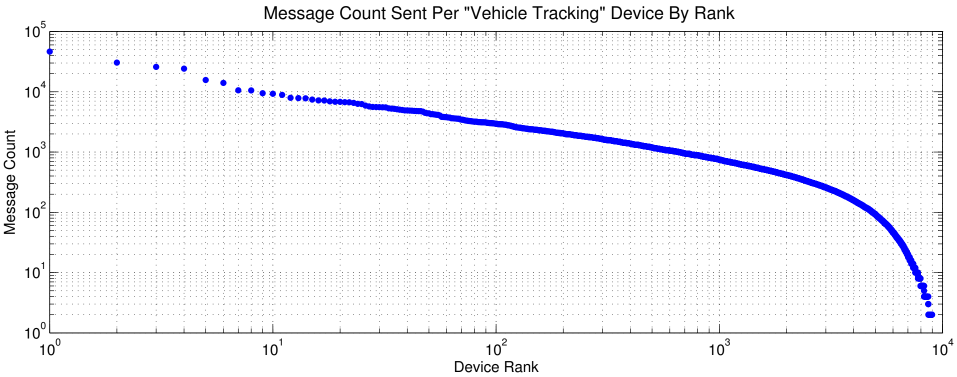


Figure 4.16. Message counts on Y for each device in the "Vehicle Tracking" application in rank order on X.

So far we have learned that devices in aggregate send varying numbers of messages at different times of day, that this can vary from day to day, and that one application can vary from another as we saw in Figure 4.8. We also easily observe that, even within the same application, devices may communicate at widely different rates. Figure 4.16 shows that a log/log scale is required to clearly capture the hyperbolic distribution of message counts across the population of devices for the "Vehicle Tracking" application.

In our data set, we have the discrete dimension of M2M device population wherein each device is uniquely identifiable, and the continuous dimension of message arrival time that we must "discretize" into time intervals, or windows. Along each of these dimensions we may then count the occurrences of n-grams for any n, with or without considering a maximum time separation of messages that identifies "motif" boundaries. In addition, we may consider the n-gram count *volumes* as they are, or as representing a probabilistic *ratio* of all messages within each sample. And lastly, we may use PCA to project the data set into a dimensional space that amplifies statistical correlations within the data.

If we limit ourselves to a single time window size of 10-minutes, n-grams of one (1) through four (4), "motif" boundaries or not for n two (2) through (4), volume vs. ratio, as well as use of PCA or not, the number of combinations available for investigation comes to $2 \times 2 \times (4 + 3) = 28$. For each of these combinations, we will need to perform k-means clustering to identify groupings of devices and time windows for many values of k, and with these groupings further determine cross-sections of the data set in both population and time resulting in 28^2k possible scenarios to consider. Thus for practical reasons we will need to prune the possibilities by analyzing the attributes of each factor to determine which lines of inquiry merit further consideration.

Since there are so many possible processing scenarios to consider, we will use the following notation when referring them:

- Original vs. PCA-projected n-gram counts

- Volume vs. Ratio of n-gram counts
- All vs Motif-based n-grams considered
- n value of the n-grams considered

Thus the original 1-gram counts without consideration for "motifs" will be labeled *OVA1*, whereas the the PCA-projected 2-gram ratios based on "motifs" will be labeled *PRM2*, and so on.

One additional scenario that we will evaluate is the alternative to using the "ratio" of n-gram counts where we use KL-Divergence instead of Eclidean distance as the metric for evaluating group membership. In this case, we will use K as label code instead of V or R.

4.2.7 Evaluating Values of k for Each Scenario

For each of our clustering scenarios, we have chosen to apply the k-means algorithm for a range of values of k from 2 through 16. In our initial experiments, we considered values of k through 32, but found that these larger values added no additional value and are not included in this dissertation.

Since k-means is not guaranteed to find the optimal clustering solution for any given value of k and since the result for any evaluation is sensitive to the random seeding of starting conditions, we chose to perform multiple evaluations for each scenario and value of k, considering the "best" solution to be that for which the SSW equation for calculating cluster distortion is minimized.

An example of this for the *ORA1* scenario of the "Vehicle Tracking" application is shown in Figure 4.17. In it, we see that for the same value of k multiple evaluations yield different cluster distortion values. What this example further highlights is a general reduction of cluster distortion as k increases, but with "local minima" found as some combinations of cluster centroid yield a "tighter" fit to the data than others.

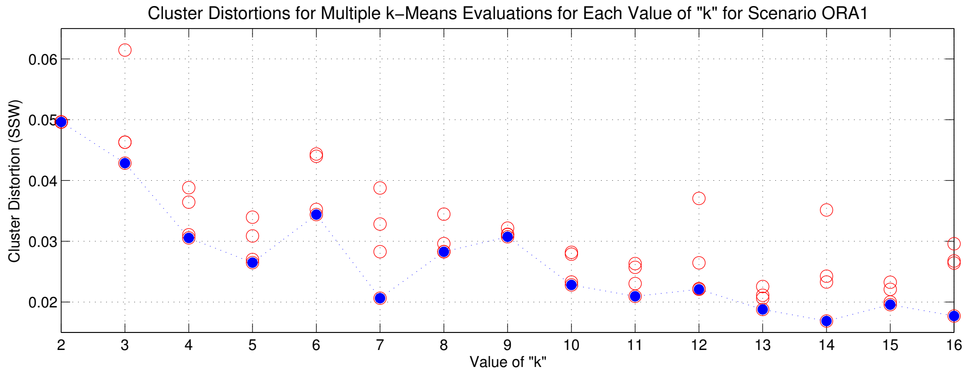


Figure 4.17. Cluster distortions computed using SSW on Y representing multiple evaluations of the k-means algorithm yielding varying results as hollow dots for each value of k from 2 through 16 on X with the "best" result for each k indicated as a solid dot; these results represent the *ORA1* scenario for the "Vehicle Tracking" application.

For simplicity, we will only use the "best" solution for each value of k in each scenario and will make no further reference to the less-optimal evaluations.

After finding the "best" clustering for a range of k values for each scenario, we are in a position to compare and contrast them to each other. In Figure 4.18 we see the cluster distortion differences among eight (8) pairs of scenarios varying original vs. PCA-projected 1- through 4-grams by volume and ratio – *OVA1-4* vs. *PVA1-4* and *ORA1-4* vs. *PRA1-4*.

One of the first things we see in these charts is that the application of PCA prior to performing k-means against the data set seems to yield different results more often for *ratio*-based (i.e., the right-hand column of charts showing *ORA1-4* vs. *PRA1-4*), rather than *volume*-based values (i.e., the left-hand column of charts showing *OVA1-4* vs. *PVA1-4*). Specifically, in the left-hand column of charts, the cluster distortion values appear almost identical for the *OVA1-4* vs. *PVA1-4* scenarios, especially for smaller-values of k, whereas in the right-hand column of charts, there the cluster distortions for *OVA1-4* vs. *PVA1-4* scenarios show more variation from one another.

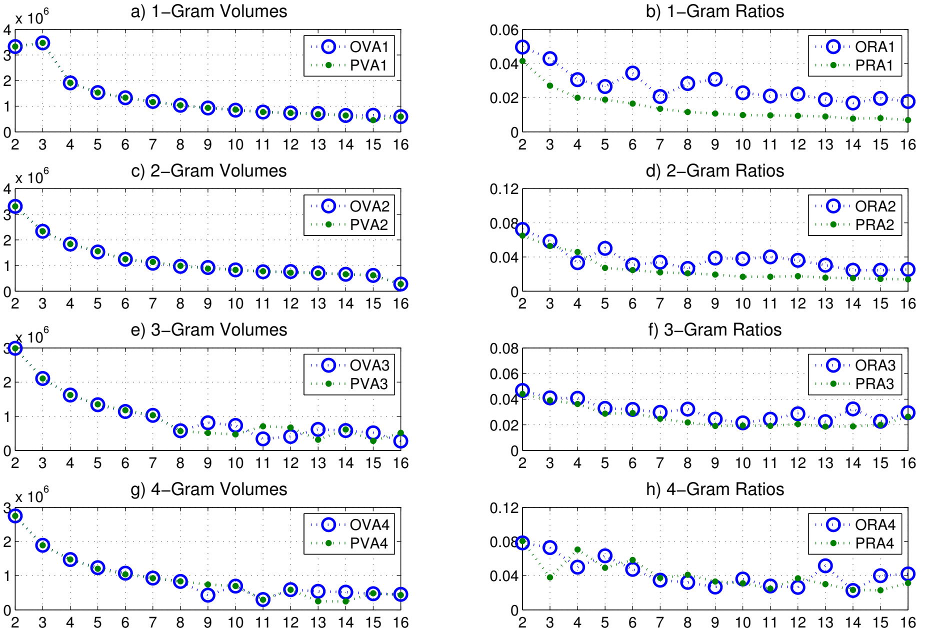


Figure 4.18. For “Vehicle Tracking” application device population clustering, charts “a,” “c,” “e,” and “g” show the Cluster Distortion (SSW) on Y for each k from 2 through 16 on X for *Volumes* of 1- through 4-grams, respectively, while charts “b,” “d,” “f,” and “h” show the Cluster Distortion (SSW) on Y for each k from 2 through 16 on X for *Ratios* of 1- through 4-grams, respectively.

To better understand why *volume*-based values seem to yield similar results, we can compare the clustering assignments of k-means with and without the prior use of PCA as seen in Figure 4.19. In each chart for values of k from 2 through 16, the charts show how the cluster assignments match between the two methods. The charts show that k of 2 through 5 result in *exactly* the same clustering results as indicated by the existence of matching clusters along the diagonal axis of the heat map. And even though for higher values of k the the cluster assignments do not exactly match between the two methods, as indicated by some cluster assignments differing such

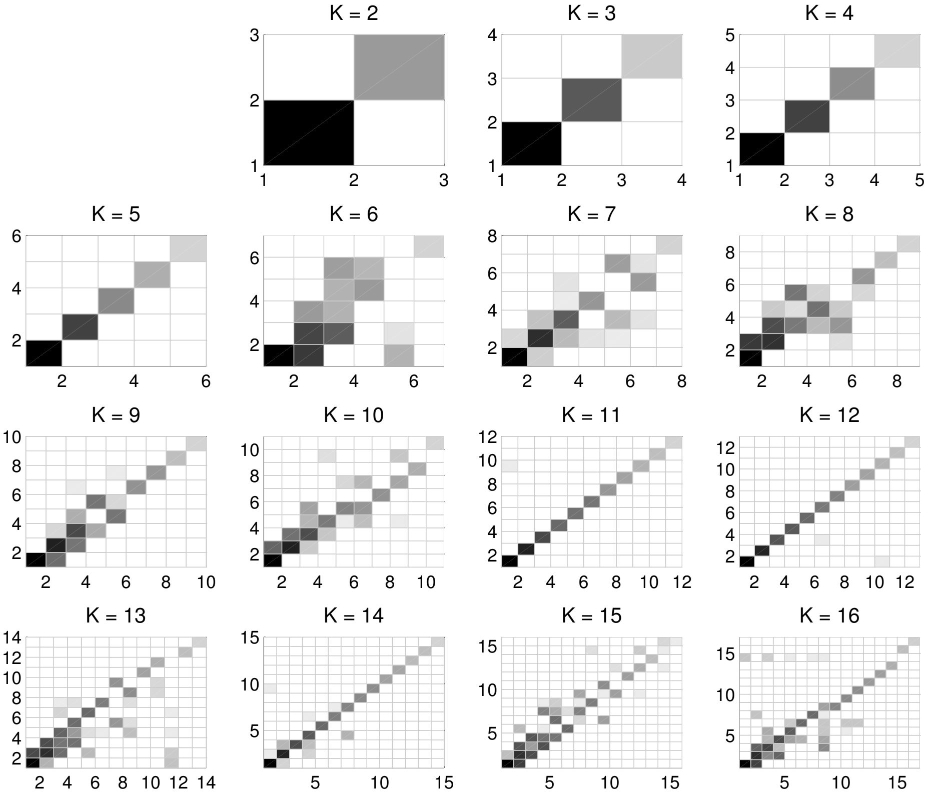


Figure 4.19. For "Vehicle Tracking" application device population 1-gram clustering by *volume*, heat maps labelled with values of k from 2 through 16 represent a comparison of the rank-ordered clustering assignments for the *OVA1* vs. *PVA1* scenarios.

that some of the assignments in one methods are spread across multiple clusters in the other, the amount of dispersal is relatively low.

By contrast when considering n-gram count *ratios* in scenarios *ORA1-4* vs. *PRA1-4*, Figure 4.20 shows a high degree of variation on the assignment of data samples to clusters with and without the application of PCA prior to k-means.

In order to understand this difference between the results using *volume*- and *ratio*-based values, we must look at the numerical properties of the two data sets.

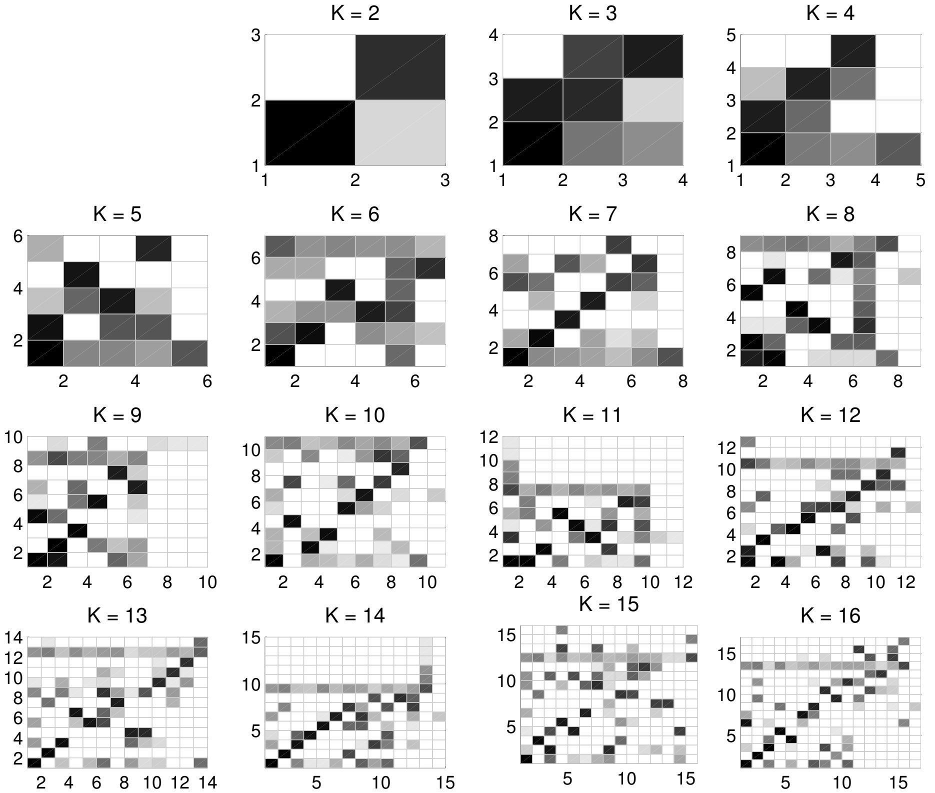


Figure 4.20. For "Vehicle Tracking" application device population 1-gram clustering by *ratio*, heat maps labelled with values of k from 2 through 16 represent a comparison of the rank-ordered clustering assignments for the *ORA1* vs. *PRA1* scenarios.

In this particular example, the raw message counts aggregated by device vary in the extreme as indicated in Figure 4.16. We also know that a very few message types dominate the overall message volume based on the hyperbolic distribution they follow as seen in Figure 4.10. Since we know that both k-means and PCA inherently are based on a least-squares, Euclidean distance internally, that they are both sensitive to "outliers," and both work best when the data distribution is Gaussian, it stands to reason that the few large-volume n-grams might dominate. We see the evidence of

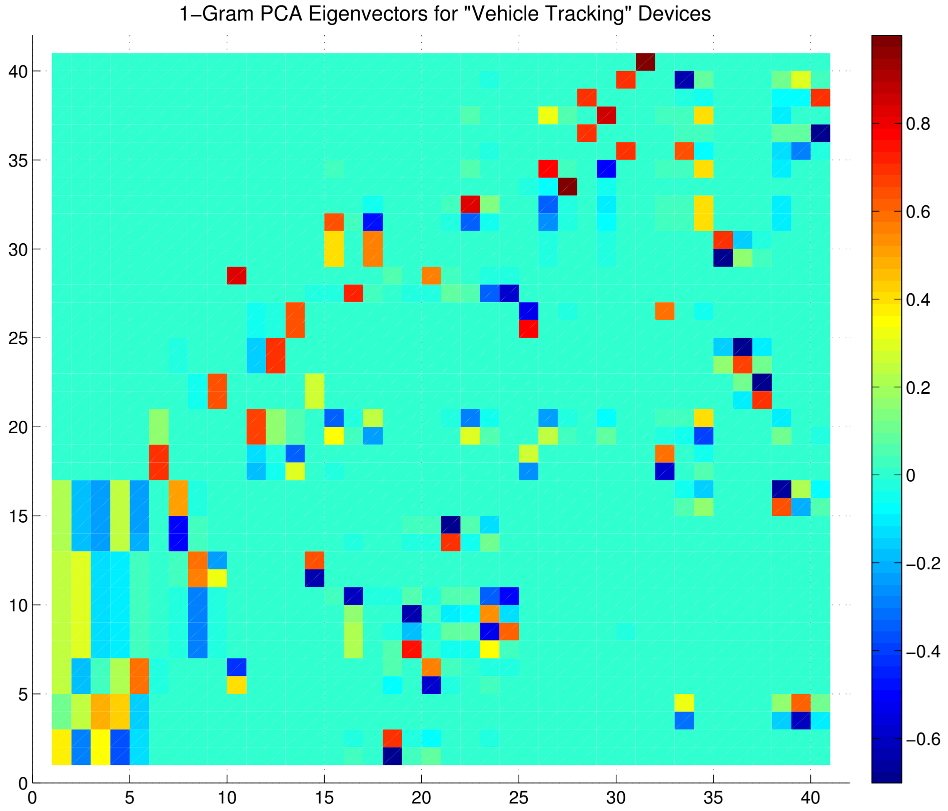


Figure 4.21. The heat map shows shows the Eigenvectors for the "Vehicle Tracking" application device population by *volume*used in the *PRA1* scenario.

this in Figure 4.21 which is heat map showing the Eigenvectors that are the result of apply PCA to the 1-gram volume counts for the "Vehicle Tracking" application. In the left-most column representing the first and most significant principle axis of the data set, we can see that the highest-ranking 1-grams dominate and with a relatively uniform contribution. As a result, projecting the original values into the resulting Eigenspace result in a data set presented to k-means that is not substantially different in the magnitude of the values used within the least-squares distance calculation that would generate a different clustering outcome.

As for the *ratios* used in this example, the *ORA1* data simply divides the count for each 1-gram by the sum of all 1-gram counts to produce the probability distribution for each device sample per Equation 2.2. By doing this, each device sample is normalized to reflect the relative relationships among the 1-gram occurrence such that a device reporting at low-frequency may have a similar range of values as one reporting at high-frequency. In this way, the extreme spread seen in the devices measured in total message counts reported from Figure 4.16 is effectively eliminated.

4.2.8 Combining Clustering Results for Population and Time

Having established clustering options for the device population and time window dimensions of our data, we are now able to use these cross-sections of the data along two independent dimensions to construct a "grid" that can partition it into smaller data sets that may reveal statistical structure hidden within the data set as a whole. Tables 4.9 & 4.10 summarized the result of processing various scenarios labeled with the designators outlined in Section 4.2.6 for source and time-window partitioning in our process by S and W for the "Vehicle Tracking" sample week.

Table 4.9. "Best k" Values for "Vehicle Tracking" Sources

Label	$k_{(n=1)}$	$k_{(n=2)}$	$k_{(n=3)}$	$k_{(n=4)}$
OVAn	4	6	8	9
ORAn	5	6	7	9
OKAn	3	5	5	6
PVAn	4	5	6	7
PRAn	4	5	9	11
OVMn	-	6	7	8
ORMn	-	6	8	9
PVMn	-	5	6	7

Table 4.10. “Best k” Values for “Vehicle Tracking” Time Windows

Label	$k_{(n=1)}$	$k_{(n=2)}$	$k_{(n=3)}$	$k_{(n=4)}$
$OVAn$	6	6	6	6
$ORAn$	6	8	8	9
$PVAn$	6	6	6	6
$PRAn$	4	4	5	7
$OVMn$	-	6	7	8
$ORMn$	-	6	7	8
$PVMn$	-	4	5	5

With we have so many scenarios for source and time clustering and so many possible k values for each, we start by looking at a specific example to gain an understanding of the what this process reveals.

The first scenario we consider is where we partition the sources S using scenario *ORM2* and time windows W using scenario *OVM2*. Most significantly, S is partitioned by the ratios of message frequency and W by absolute message counts. From Tables 4.9 & 4.10 we select $k_\Gamma = 6$ and $k_\Phi = 6$. Figure 4.22 shows each intersect χ in a grid with each γ in a column on the right and each ϕ on the bottom. Each cell of the grid shows the mean, standard deviation, and min/max envelope of the samples in the partitions.

What we can see in Figure 4.22 that γ_1, γ_2, γ_3, and γ_6[1] are populations of devices that each show very specific message usage profiles at relatively low frequency. Conversely populations represented by γ_4 and γ_5 show far higher levels of message counts. The actual number of devices assigned to each γ is shown in Table 4.11.

[1] Due to limitations in the character set available for the charting package, the labels for γ and ϕ partitions will use S and W, respectively, instead.

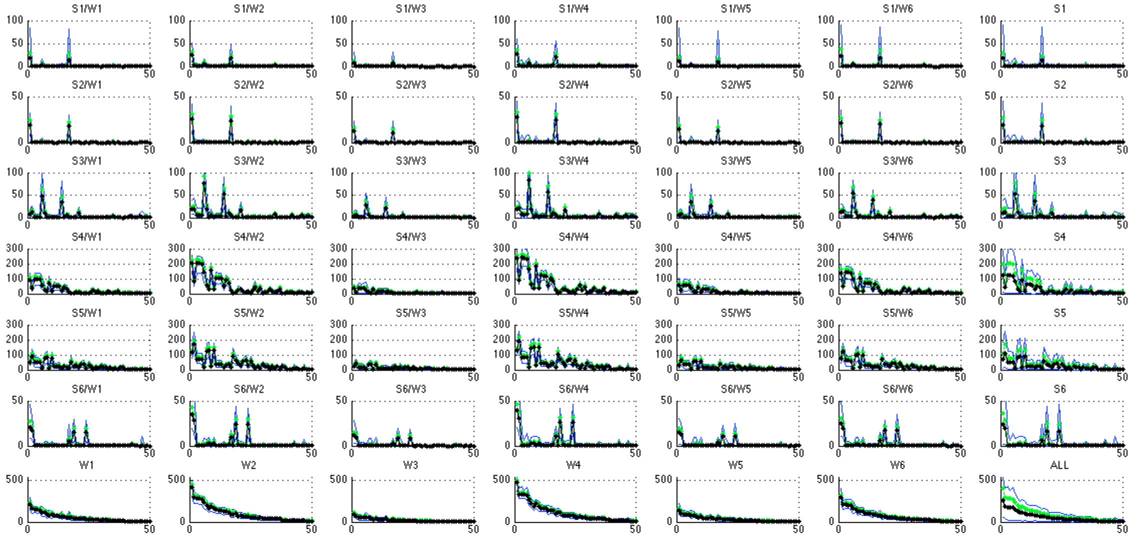


Figure 4.22. Each chart shows the ***absolute counts*** of the "Vehicle Tracking" SS7 $S_{ORM2} \times W_{OVM2}$ partitioning where $k_\Gamma = 6$ and $k_\Phi = 6$ with the mean in black, standard deviation in green, and min/max envelope in blue for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

By using *OVM2* as the scenario for partitioning W, we see that each column in Figure 4.22 corresponds to a ϕ partition that shows clear variation in absolute counts and likewise the corresponding χ partitions for γ_4 and γ_5 show correlation in rise and fall of total counts, while the γ_1, γ_2, γ_3, and γ_6 message counts remain relatively constant for all ϕ partitions.

What we observe about these constant-reporting device populations is that they each communicate using a very small number of messages repetitively. For example, source partitions γ_1 and γ_2 each send almost equal occurrence counts of 2-grams `ISH56-RSH56` and `IVH07-RVH07` corresponding to separate requests for "Send Authentication Info" and "Insert Subscriber Data." Given our use of "motifs," the absence of the occurrence of `RSH56-RVH07` 2-gram tells us that these message sequences occur at times separated by more than Δm (that is, 60 seconds) and do not occur in

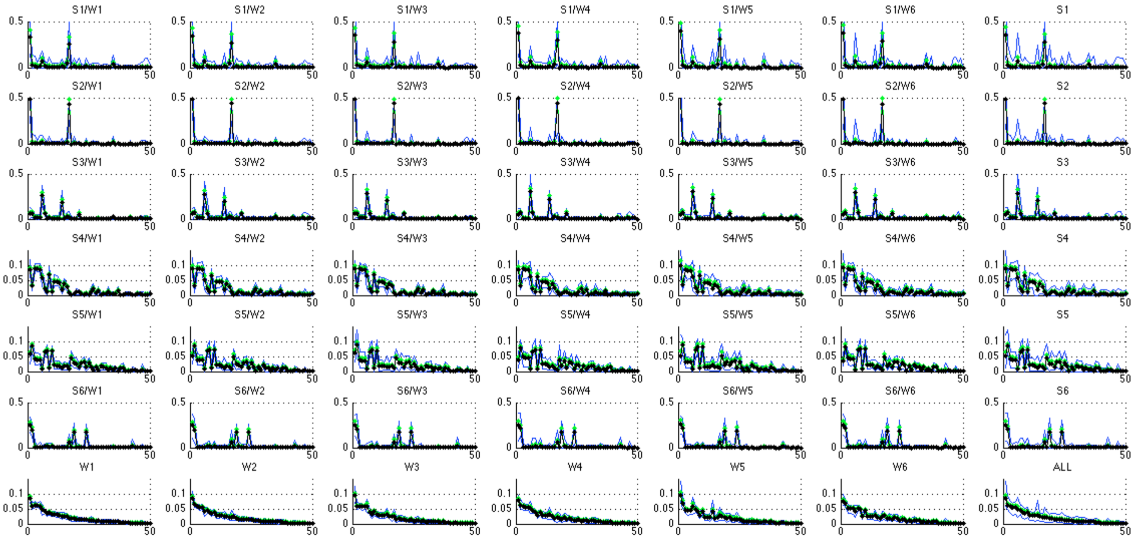


Figure 4.23. Each chart shows the ***ratios*** of the "Vehicle Tracking" SS7 $S_{ORM2} \times W_{ORM2}$ partitioning where $k_\Gamma = 6$ and $k_\Phi = 6$ with the mean in black, standard deviation in green, and min/max envelope in blue for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

a single burst. This kind of information is useful to anyone who needs to understand the meaning of these occurrence patterns.

Another way to create X intersects is using the the *ORM2* as the scenario for partitioning W. This pairs a partitioning of source and time window by the same scenario using message occurrence ratios rather than absolute counts. Figure 4.23 shows this breakdown and uses a similar statistical summary of each partition of mean, standard deviation, and min/max envelope, but with the Y-axis showing the probabilities of each symbol instead of the absolute counts. In this case, the ϕ partitions of W, and therefore the χ intersects of both S and W, are not related to rise and fall of data over time, but to the "mix" of messages occurring at different times.

In Figure 4.23 we still see similar statistics for χ partitions corresponding to each γ, but do observe small variations across different ϕ groupings as different 2-

Table 4.11. "Vehicle Tracking" SS7 Partitioning Statistics Using $k_\Gamma = 6$

Partition	Devices (D)	Messages (M)	Rate (M / D)
γ_1	240	54,781	228
γ_2	4	40,678	10,169
γ_3	30	195,154	6,505
γ_4	45	1,411,645	31,369
γ_5	7,963	1,283,979	161
γ_6	690	96,750	140

gram occurrences change in their relative contribution to the ratios. Looking closely at the specific message types that vary from ϕ to ϕ will inform the analyst as to what kind of behavioral differences exist from time to time.

4.2.9 Identification of Anomalies

At one level, the outcome of the intersect partitioning may show an analyst interested in the system that ths behavior patterns of certain device populations is, in and of itself, anomalous. For example, Table 4.11 highlights that the γ_2, γ_3, and γ_4 partitions have dramatically higher reporting rates that may represent aberrant behavior. In fact, this is an example of how valuable information is derived at all steps of our process.

But our method makes no judgements as to the "correctness" of the behavior identified. It simply discovers the existence of differences. So from the standpoint of process, an anomaly is simply the occurrence of a sample in our data set that falls outside some norm.

The most straightforward norm to consider is the distance of a sample from the centroids of either the marginal – γ or ϕ – and intersect – χ – partitions to which it

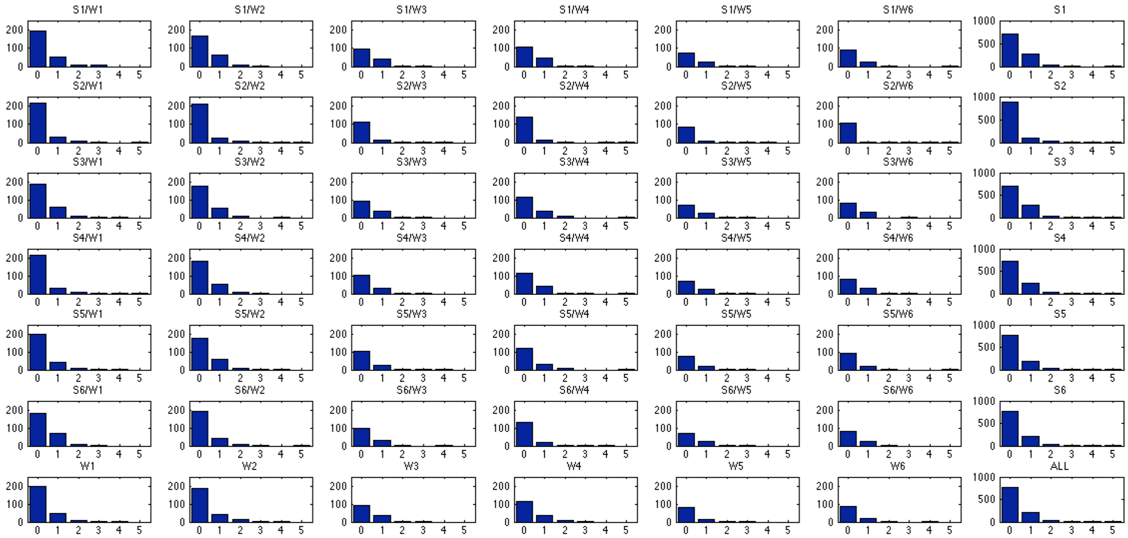


Figure 4.24. Each histogram shows the number of "Vehicle Tracking" SS7 samples with standard deviations of the distances from the mean from 0 through 5 for the $S_{ORM2} \times W_{ORM2}$ partitioning where $k_\Gamma = 6$ and $k_\Phi = 6$ for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

is assigned. A simple way to status the distance of the samples is by assigning each to a "bin" defined by the number of standard deviations it is away from the centroid. Figure 4.24 shows an example of our partition grid visualization where each cell of is a corresponding histogram of these bins.

We may further simplify the visualization of these histograms into simple pie charts that show "good" as green for bins ≤ 2, "bad" as yellow for bins > 2 and ≤ 4, and "worse" as red for bins > 4.

One of the types of anomalies that we can detect with this process we called a "migration anomaly" where the behavior of a source changes to appear like another type of source. An anomaly of this type is shown in Figure 4.26. Using the "baseline" method for analyzing newly arriving time windows of size Δw, we see a change in a large number of the devices in source partition γ_4 that began to communicate

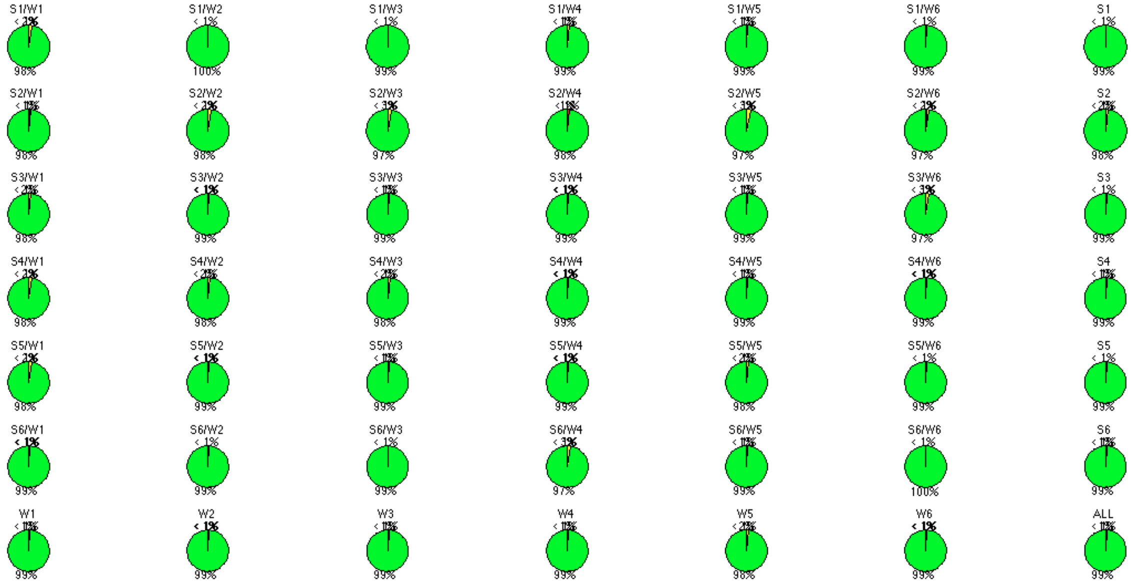


Figure 4.25. Each pie chart shows the percent of "Vehicle Tracking" SS7 samples with standard deviations of the distances from the mean as "good" (≤ 2) "bad" ($> 2 \leq 4$), and "worse" (> 4) for the $S_{ORM2} \times W_{ORM2}$ partitioning where $k_\Gamma = 6$ and $k_\Phi = 6$ for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

using the same behavior as γ_1 after 1, 4, and 24 hours. In the figure, we see how, as the duration of the migration anomaly persists, the impact shows up within a larger number of χ partitions where $\chi_{*,4} = (\phi_*, \gamma_4)$ using both the pie and histogram visualizations.

Note that the "migration anomaly" represented here may, in fact, be a desirable outcome since the high-reporting γ_4 devices reporting at the low levels of the γ_1 devices represents a drop in message traffic levels. Again, our method does not truly consider changes detected as "good, bad, worse" but something more like "similar" vs. "different."

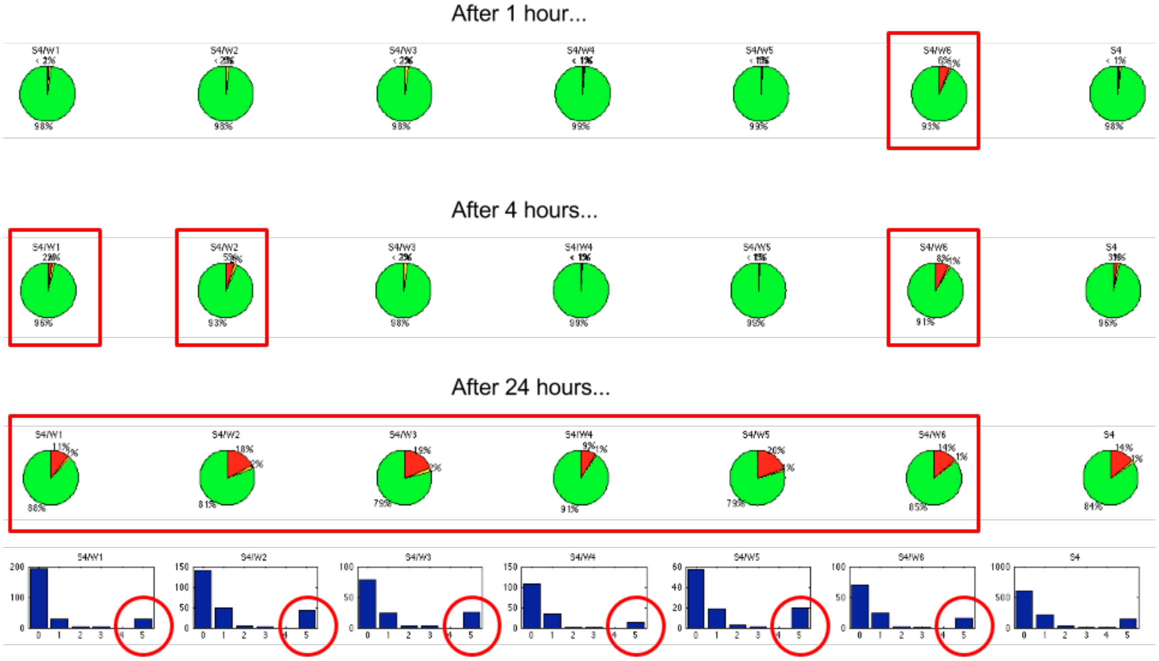


Figure 4.26. The figure shows the “Vehicle Tracking” SS7 $S_{ORM2} \times W_{ORM2}$ partitioning pie charts for $k_\Gamma = 6$ and $k_\Phi = 6$ where a “migration” of devices in group γ_4 begin behaving like γ_1 after 1, 4, and 24 hours.

4.2.10 Summary of SS7 Experimental Learnings

As shown in Tables 4.9 & 4.10, we performed partitioning for 28 source scenarios and 24 time window scenarios. making for a total of 672 possible partitioning combinations. Of these, we found *OVM2* and *ORM2* of most utility from a qualitative perspective – hence the prominence of these in our examples. That said, several other combinations seemed to have merit, and under the right circumstances could easily have been chosen instead. The following are some observations regarding the other scenario options.

- **PCA** – PCA featured heavily in our analysis, especially for the $PV{**}$ scenarios. Since the cost of clustering has a time complexity of that increases proportionally to the number of dimensions, using a subset of the dimension of our data sets after projecting them into an *eigenspace* defined by the PCA process produced clusters and could "detect anomalies" within the *eigenspace*, but this made mapping the results to the actual n-gram messaging domain difficult to interpret.

As as alternative, we used the PCA coefficients matrix such as in Figure 4.13 to identify those n-gram columns that were represented with non-zero contribution in the first several *eigenvectors* to limit the n-grams considered for subsequent processing, assigning all other n-grams to "other" as outlined in Section 3.2. This approach has the advantage of reducing the time complexity of the process and shows promise for practical applications.

- **KK-means and KL-Divergence** – As stated, this method is only possible when dealing the the $*R**$ scenarios as it requires the samples to be probability distributions that have meaning as input to the KL-Divergence function.

  In our experiments, we were not able to get this method to converge with any of our data sets for the W dimension. The reason for this could be that the overall distribution of n-grams over time is relatively uniform that it was difficult for the algorithm to converge on a solution since cluster membership would continually shift between iterations.

  We were able to get reasonable convergence after many evaluations for S, but only for lower values of k. Since all other scenarios ultimately rely on Euclidean distance and statistics like standard deviation, using KL-Divergence made comparing results difficult. In general, this approach has promise and is worth further study.

- **All n-Grams** – Given the clear "bursty" arrival of the messages in this domain, eliminating those n-grams with internal time delays $> \Delta m$ seemed to make sense. In performing the partitioning calculations both with all vs. "motif"

n-grams, we did see similar results, so perhaps the most significant advantage to "motif" over all n-grams comes through the information provided to analysts trying to interpret results. Also, since all n-grams generally require the collection of 15-20% more symbols, their elimination has a modest improvement on process performance. As a result, we see no reason to pursue both all vs. "motif" n-gram scenarios further for this domain.

4.3 Study: Cellular Wireless Connection Management

Now that we have explored the SS7 protocol layer, we turn our attention to another layer of the cellular wireless communications stack handling connection management with similar linguistic properties called "General Packet Radio Service" (GPRS) [71]. This layer is used by M2M applications that rely on IP-based data communications and the remote devices interact with different network elements than those used in signalling, and use a different "language" for communicating with them.

Since we went into much detail using SS7 to explain the nuances of the our process at each stage, we will be able to move much more quickly with GPRS.

The first thing we notice when working with GPRS is that, as with SS7, there are hundreds of message types in the GPRS "alphabet" as defined by the "GPRS Tunnelling Protocol" (GTP). However, in practice we experience far fewer unique occurrences for the same population of M2M devices than we saw using SS7.

Table 4.12. Comparison of SS7 and GPRS Metrics

Attribute	SS7 (S)	GPRS (G)	Percent (G / S)
Unique Devices	1,651,967	754,875	46%
Total Messages	804,302,527	45,785,702	6%
Messages Per Device	487	60	12%
Message Types	72	9	14%

Table 4.12 compares some of the metrics that differ between SS7 and GPRS for a given sample week for our experimental data sources. The devices are the same with both SS7 and GPRS, but the GPRS count is smaller since only a subset of the devices actually use the GPRS services for IP communications. Not only does GPRS experience far fewer message types operationally as compared with SS7, messages are sent less frequently.

Table 4.13. GPRS Message Types and Occurrences

Label	Occurrences	Description
0x10:128	19,939,221	Create PDP Context - OK
0x14:128	19,428,593	Delete PDP Context - OK
0x12:128	4,022,993	Update PDP Context - OK
0x10:222	2,221,451	Create PDP Context - Access Denied
0x14:192	60,579	Delete PDP Context - Not Found
0x10:211	29,932	Create PDP Context - No Address Available
0x12:209	10,643	Update PDP Context - Auth Failure
0x10:209	297	Create PDP Context - Auth Failure
0x10:199	1	Create PDP Context - No Resources Available

Table 4.13 shows the nine (9) message types that typically occur, with only four (4) logical operations paired with response code of either success or one of several possible errors. Since GPRS provides connection (PDP context) management for IP packets sent and received by mobile devices, the primary messages we see are *create*, *update*, and *delete* operations that establish the PDP context that must exist to allow an IP packet to be transmitted. The occurrence counts in Table 4.13 show the dramatic difference in frequency of these messages consistent with “Zipf’s Law” as we’ve seen elsewhere. Along with these messages, 70% of the total GPRS message traffic are the actual application-level IP packets. So this analysis looks only at the “control” messages that enable those application packets.

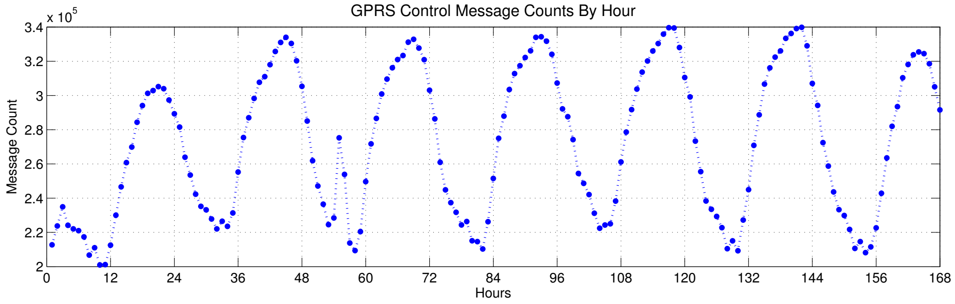


Figure 4.27. The chart shows the total GPRS message count on Y for every hour of the target week on X for all devices in the data set.

As with SS7, the arrival of messages shows strong diurnal and other periodic trends which we can see in Figure 4.27. In addition, we see a notable spike in traffic in the third trough from the left.

Moving on the n-gram analysis, Table 4.14 shows the slow growth of actual unique occurrences as compared to the exponential maximum that we have seen before.

Table 4.14. Possible vs. Actual n-Gram Occurrences for GPRS

N	$\|\|A\|\|^N$	=	Possible (P)	Actual (A)	Percent (A/P)
1	9^1	=	9	9	100.0%
2	9^2	=	81	51	63.0%
3	9^3	=	729	197	27.0%
4	9^4	=	6,561	564	8.6%

Due to the small census of n-grams at any level, we set aside an analysis of PCA since its ability to reduce dimensionality is not required. Since the control messages for connection management are intended to be infrequent, we do not use the notion of "motifs" either. Thus we focus only on the $OVAn$ and $ORAn$ scenarios for partitioning.

In applying k-means to S, we quickly find that it consistently fails to converge when $k_\Gamma = 5$. Furthermore, when applying our cost function from Equation 3.7 $k_\Gamma = 4$ consistently emerges as the choice. For W we evaluate k_Φ from 2 through 16, we settle on $k_\Phi = 6$.

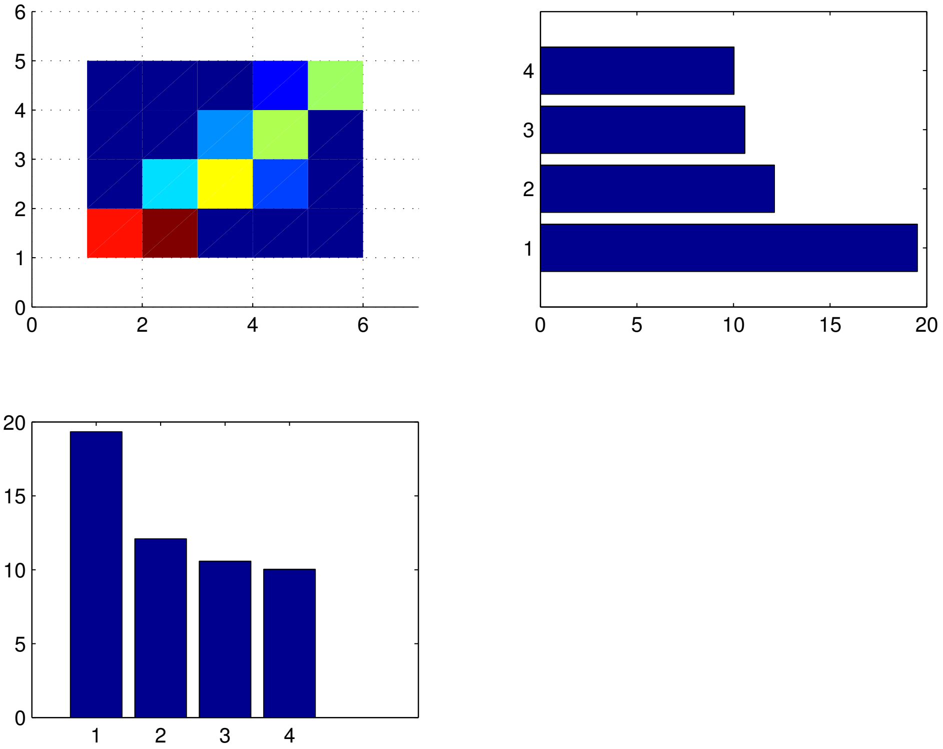


Figure 4.28. The top-left heat map shows matching cluster assignments for GPRS 1-gram γ's on X and 4-gram γ's on Y. The bottom bar chart shows the number of assignments for each of the 1-gram γ's below the X axis. The right-hand bar chart shows the number of assignments for each of the 4-gram γ's aligned with the Y axis.

When considering what level of n-gram to use for our analysis, we compared the results of the partitioning for each and found that they were remarkably consistent from levels 1 through 4. Figure 4.28 shows a visual comparison of the cluster assignments using a heat map that shows the similarity of each cluster assignment for 1-grams and 4-grams when $k_\Gamma = 4$. The high similarity is reflected in the fact that most of the assignments follow the diagonal. As a result, we choose to simply use 1-grams for our further analysis, since additional n-grams provide little no additional criteria for partitioning.

Table 4.15. GPRS Partitioning Statistics Using $k_\Gamma = 4$

Partition	Devices (D)	Messages (M)	Rate (M / D)
γ_1	1,537	2,448,211	1,592
γ_2	4,427	3,713,898	838
γ_3	1,045	5,348,473	5,118
γ_4	747,866	34,275,120	45

Table 4.15 shows that most of the devices cluster into the γ_4 partition and have a relatively low per-device reporting rate. The other smaller clusters in γ_1 through γ_3 have much higher reporting rates. These clusters could, in and of themselves, represent anomalous populations from an external perspective, but in this case, they are simply identified as what exists.

Using a classification matrix for $S_{ORA1} \times W_{OVA1}$ showing absolute counts in Figure 4.29, we can see clear similarity to the absolute counts of (ϕ_*, γ_4) due to its overwhelming size in terms of message counts seen in Table 4.15. The variations found in the other χ are overwhelmed and have very little impact on the totals.

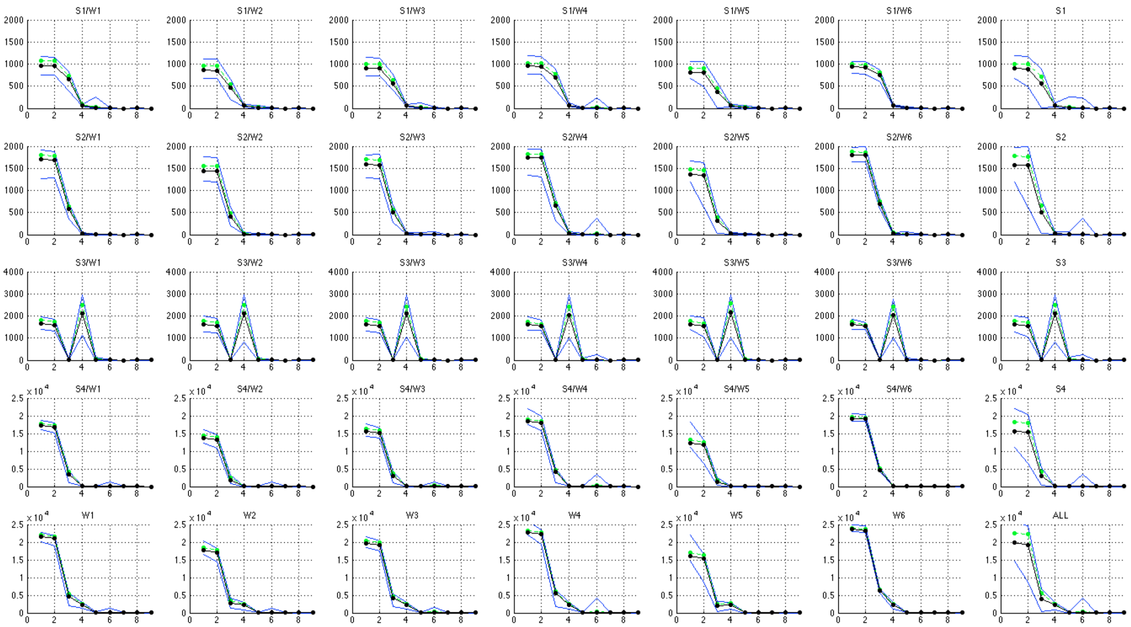


Figure 4.29. Each chart shows the ***absolute counts*** of the GPRS $S_{ORA1} \times W_{OVA1}$ partitioning where $k_\Gamma = 4$ and $k_\Phi = 6$ with the mean in black, standard deviation in green, and min/max envelope in blue for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

Figure 4.30 shows the $S_{ORA1} \times W_{ORA1}$ partitioning statistics for ratios, followed by the histogram and pie status grids for the same in Figures 4.31 and 4.32, respectively.

Were we to see a material change in the flow of any of the source partitions as we did with the "migration anomaly" in the SS7 study, we could likewise see similar changes in the histograms and pies. But in this study, we show how we can detect other classes of anomalies. In particular, we show in Figure 4.33 a strip chart that shows an indication of the message volume for each γ source partition with W on the X axis and the values of the strip chart also showing green to indicate OK message responses and red for error responses. What we see is that errors are frequent for the

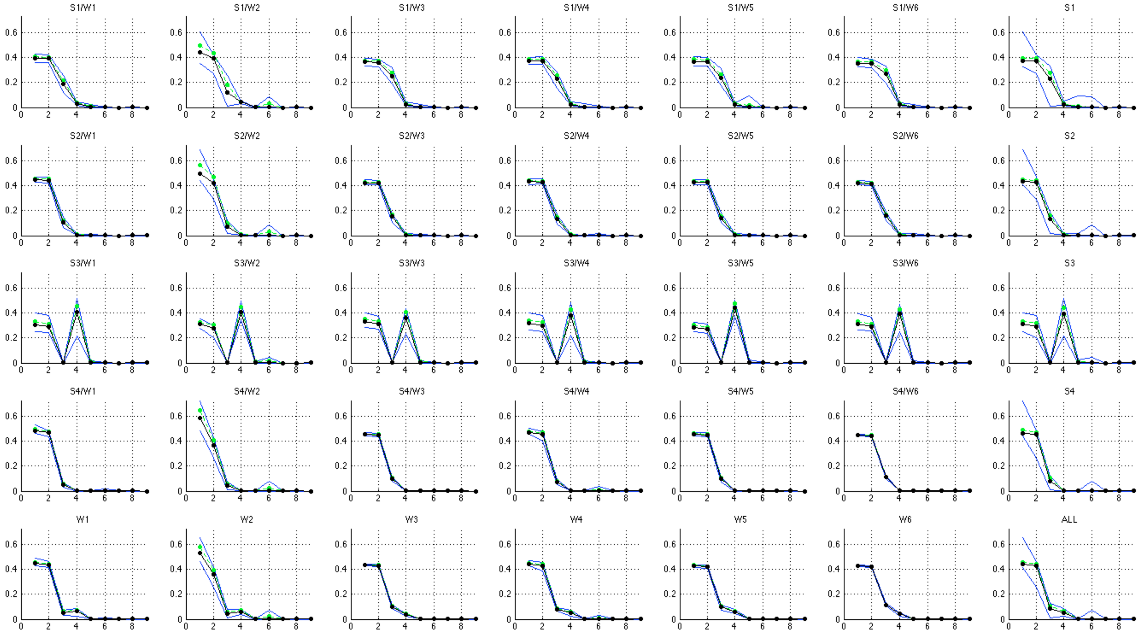


Figure 4.30. Each chart shows the ***ratios*** of the GPRS $S_{ORA1} \times W_{ORA1}$ partitioning where $k_\Gamma = 4$ and $k_\Phi = 6$ with the mean in black, standard deviation in green, and min/max envelope in blue for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

γ_3 partition, but there is also a clear error event occurring across the other partitions near minute 6000 on the time line.

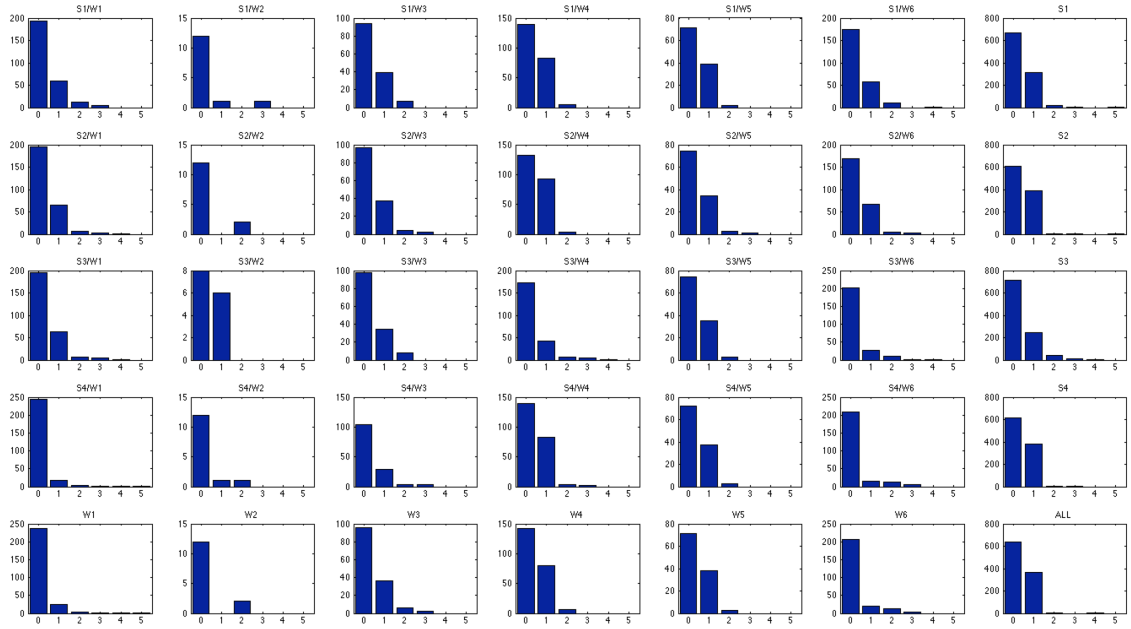


Figure 4.31. Each histogram shows the number of GPRS samples with standard deviations of the distances from the mean from 0 through 5 for the $S_{ORA1} \times W_{ORA1}$ partitioning where $k_\Gamma = 4$ and $k_\Phi = 6$ for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

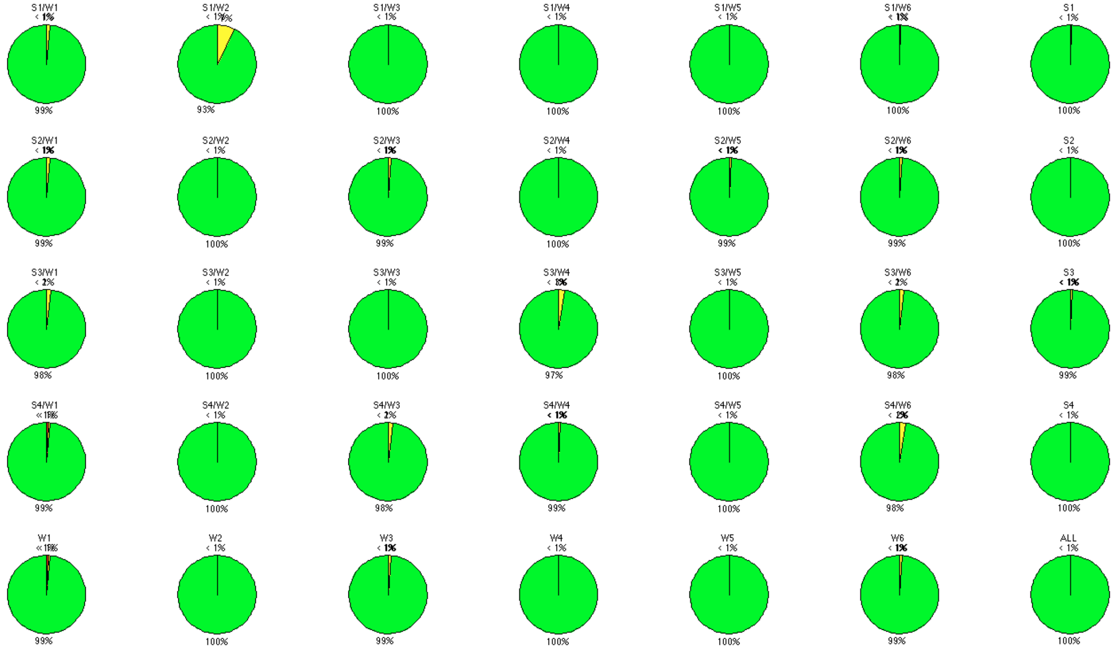


Figure 4.32. Each pie chart shows the percent of GPRS samples with standard deviations of the distances from the mean as "good" (≤ 2) "bad" ($> 2 \leq 4$), and "worse" (> 4) for the $S_{ORA1} \times W_{ORA1}$ partitioning where $k_\Gamma = 4$ and $k_\Phi = 6$ for each intersect S_y/W_x with source summaries S_y in the right-hand column, window summaries W_x in the bottom row, and a summary for the entire data set in the bottom-right corner.

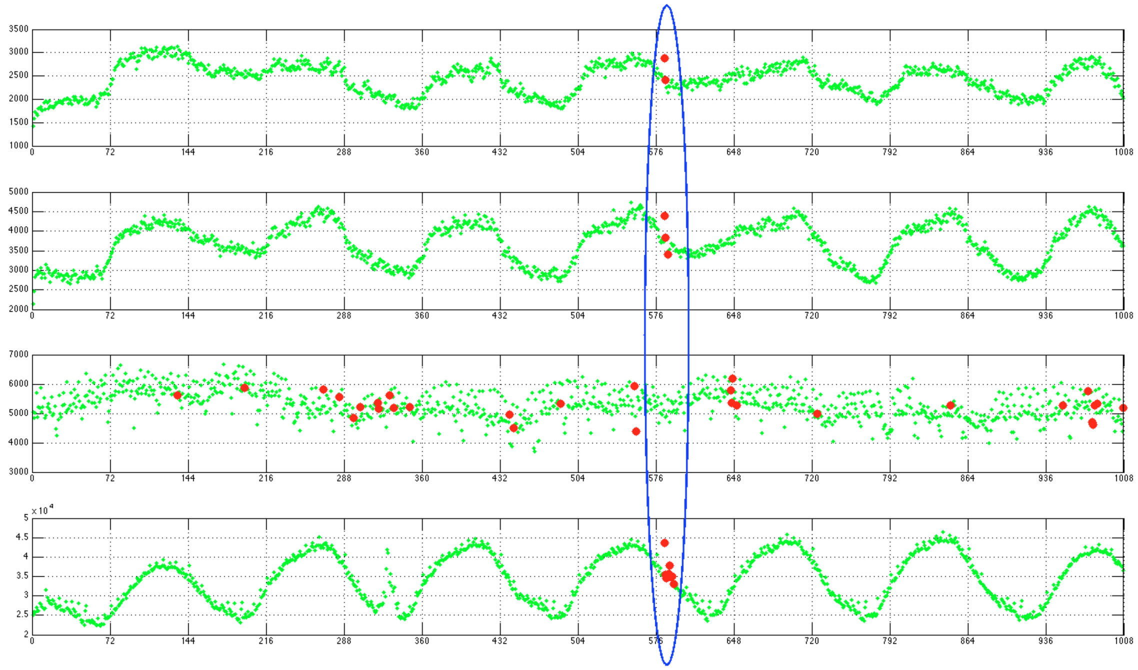


Figure 4.33. The strip chart shows GPRS samples on the time line broken out for each γ with "good = green" for OK responses and "bad = red" for error responses to the various PDP context operations for the $S_{ORA1} \times W_{ORA1}$ partitioning where $k_\Gamma = 4$ and $k_\Phi = 6$; the blue oval highlights an event occurring across all source partitions.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Summary of Findings

To review, we have identified a class of multidimensional time-series that exhibit "bursty," asynchronous data arrival from perhaps very large numbers of sources over a common channel in which internal structure is suspected to exist, but for which clear means for identifying that structure do not.

We have laid out the mathematical notation and selected several mechanisms as building blocks for a unique process to decompose such time-series into groupings defined by similar information source, on the one hand, and aggregate time windows, on the other, and combine these groupings to partition our time-series into cross-sections whose statistical properties reveal previously-unknown attributes that illuminate differences in behavior of the information sources that may occur at different times.

Using these source/time partitions, we have identified anomalous sources and time periods that were previously undetectable. We have demonstrated this with several contrived and real-world data sources, each with different scale and dimension, highlighting the adaptability of the approach. In addition, we have several ways to visualize anomalies, as well as basic behavioral properties, that are of value to anyone trying to understand the systems being studied.

Our unique contributions here are, first and foremost, the characterization and analysis process itself which represents a new method of discovering hidden internal aspects of a systems behavior. Included in this process are several additional unique contributions. We have applied the notion of n-grams and a linguistic analysis to time-series not previously encountered in the literature, and in addition, have introduced a novel "motif" concept to identify boundaries of symbol sequences that would otherwise not exist. Finally, we have combined "marginal" clustering along several dimensions to form "intersects" of time-series that are key to revealing the hidden behavior of the information sources of the systems being studied.

5.2 Potential Target Systems

While we have shown how our process applies to several real-world systems, there are others with quite different qualities that we can envision.

Once such target system would be Twitter. This H2H system is characterized by millions of end-users (think S) broadcasting short messages on no particular schedule. As we saw with M2M systems, some sources may frequently send message while others may not. Given that humans follow diurnal patterns, we would expect much temporal variability in message rates as well (think W). Several lines of inquiry such as in [72, 73, 74] are underway to classify "tweets" using a discrete dictionary of sentiments, emotions, and the like (think A) that suggest that Twitter might well be a good fit for our methods.

Another target system would be location-based applications. While one of the applications in our experiments is a vehicle tracking application, we were investigating the behavior of this system at a level where geographic locations were not being considered. But we know that such applications contain histories of latitude/longitude coordinates for vehicles (think S) reporting at different times (think W), some while in motion and others at rest. We can imagine that some locations, such frequent overnight stops or well known "geofences," could be converted to a discrete set of symbols (think A). As such, we have all the elements necessary to apply our method.

5.3 Additional Future Work

In the future, there are many more avenues that we may pursue with this approach. Our process is highly parametrized, and offers many different options for conditioning the data for a variety of outcomes. We explored several of the parameter combinations, but much more could be done to investigate them. It is also possible to introduce alternative mechanisms to the ones in our studies. For example, alternatives to k-means as a clustering tool could be considered. It is also possible that more "marginal" dimensions other than source and aggregate time window could be brought to bear for a further partitioning of time-series into cross-sections that might reveal additional structure within the data.

REFERENCES

[1] E. de Argaez, “WORLD INTERNET USAGE AND POPULATION STATISTICS,” 2013. [Online]. Available: http://www.internetworldstats.com/stats.htm

[2] D. Evans, “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything,” *Cisco IBSG*, no. April, pp. 1–11, 2011.

[3] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, “A First Look at Cellular Machine-to-Machine Traffic Large Scale Measurement and Characterization,” in *SIGMETRICS'12*, 2012, pp. 1–12.

[4] Y. Chen and W. Wang, “Machine-to-Machine Communication in LTE-A,” *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pp. 1–4, Sept. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5594218

[5] M. J. Booysen, J. S. Gilmore, S. Zeadally, and G. V. Rooyen, “Machine-to-Machine (M2M) Communications in Vehicular Networks,” *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol. X, no. X, pp. 1–21, 2011.

[6] C. Wietfeld, H. Georg, S. Gr, C. Lewandowski, and J. Schmutzler, “Wireless M2M Communication Networks for Smart Grid Applications,” in *European Wireless 2011*, 2011, pp. 275–281.

[7] S. Abdul Salam, S. Mahmud, G. Khan, and H. S. Al-Raweshidy, “M2M communication in Smart Grids: Implementation scenarios and performance analysis,” *2012 IEEE Wireless Communications and Networking Conference*

Workshops (WCNCW), vol. 1, pp. 142–147, Apr. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215478

[8] ETSI Technical Committee Machine-to-Machine Communications (M2M), "Machine-to-Machine Communications (M2M): Functional Architecture ETSI TS 102-690 V1.1.1 (2011-10)," 2011.

[9] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2M : From Mobile to Embedded Internet," *IEEE Communications Magazine*, no. April, pp. 36–43, 2011.

[10] M. Schneps-Schneppe and D. Namiot, "Open API for M2M Applications: What is Next? Current state and development proposals," in *AICT 2012: The Eighth Advanced International Conference on Telecommunications*, no. c, 2012, pp. 18–23.

[11] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M Networks : Architectures , Standards , and QoS Improvement," *IEEE Communications Magazine*, no. April, pp. 44–52, 2011.

[12] R. Liu, W. Wu, H. Zhu, and D. Yang, "Network," in *7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2011, pp. 1–5.

[13] S. Poua and D. Giljeviü, "Machine to machine (M2M) communication impacts on mobile network capacity and behaviour," in *MIPRO*, 2012, pp. 607–611.

[14] D. Helbing and S. Balietti, "How to Do Agent-Based Simulations in the Future : From Modeling Social Mechanisms to Emergent Phenomena and Interactive Systems Design," 2011.

[15] D. Drajic, S. Krco, I. Tomic, P. Svoboda, M. Popovic, N. Nikaein, and N. Zeljkovic, "Traffic generation application for simulating online games and M2M applications via wireless networks," *2012*

9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), pp. 167–174, Jan. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6152224

[16] M. Beale, "Future challenges in efficiently supporting M2M in the LTE standards," *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 186–190, Apr. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215486

[17] C. Ide, B. Dusza, M. Putzke, C. Muller, and C. Wietfeld, "Influence of M2M communication on the physical resource utilization of LTE," *Wireless Telecommunications Symposium 2012*, pp. 1–6, Apr. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6266084

[18] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, "M2M Scheduling Over LTE," *IEEE Vehicular Technology Magazaine*, no. September, pp. 34–39, 2012.

[19] F. Andreini, F. Crisciani, C. Cicconetti, and R. Mambrini, "Context-aware location in the Internet of Things," *2010 IEEE Globecom Workshops*, pp. 300–304, Dec. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5700330

[20] S.-y. Lien, K.-c. Chen, and Y. Lin, "Toward Ubiquitous Massive Accesses in 3GPP Machine-to-Machine Communications," *IEEE Communications Magazine*, no. April, pp. 66–74, 2011.

[21] A. Lo, S. Member, Y. W. Law, and M. Jacobsson, "Enhanced LTE-Advanced Random-Access Mechanism for Massive Machine-to-Machine (M2M) Communications," in *27th Meeting of Wireless World Research Forum*, 2011, pp. 1–7.

[22] P. Makris, D. N. Skoutas, N. Nomikos, D. Vouyioukas, and C. Skianis, " Context-Aware Backhaul Management Solution for combined H2H and M2M traffic," in

IEEE International Conference on Computer, Information and Telecommunication Systems (CITS), 2013, pp. 1–5.

[23] F. M. Zanzotto, M. Pennacchiotti, and K. Tsioutsiouliklis, “Linguistic Redundancy in Twitter,” in *2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 659–669.

[24] C. Tseng, N. Patel, H. Paranjape, T. Y. Lin, and S. Teoh, “Classifying Twitter Data with Naive Bayes Classifier,” in *2012 IEEE International Conference on Granular Computing Classifying*, 2012, pp. 2–6.

[25] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, no. July 1928, pp. 379–423, 1948.

[26] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[27] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=61115

[28] W. B. Cavnar, J. M. Trenkle, and A. A. Mi, “N-Gram-Based Text Categorization,” in *SDAIR-94: 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161—-175.

[29] L. Khreisat, M. Ave, and M. Nj, “Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study,” in *DMIN 2006: International Conference on Data Mining*, 2006, pp. 78–82.

[30] A. Ito and M. Kohda, “Language Modeling by String Pattern N-gram for Japanese Speech Recognition,” in *Fourth International Conference on Spoken Language*, 1996, pp. 4–7.

[31] J. Houvardas and E. Stamatatos, “N-Gram Feature Selection for Authorship Identification,” in *Artificial Intelligence: Methodology, Systems, and Applications.* Springer, 2006, pp. 77–86.

[32] M. Ganapathiraju, D. Weisser, R. Rosenfeld, J. Carbonell, and R. Reddy, “Comparative n-gram analysis of whole-genome protein sequences,” in *HLT ’02 Proceedings of the second international conference on Human Language Technology Research*, 2002, pp. 76–81.

[33] Z. Xing, J. Pei, and E. Keogh, “A brief survey on sequence classification,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 1, p. 40, Nov. 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1882471.1882478

[34] S. Bergsma, E. Pitler, and D. Lin, “Creating Robust Supervised Classifiers via Web-Scale N-gram Data,” in *ACL ’10 Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 865–874.

[35] D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, K. Dalwani, and S. Narsale, “New Tools for Web-Scale N-grams,” in *LREC 2010: International Conference on Language Resources and Evaluation*, 2010, pp. 2221–2227.

[36] A. Juvonen and T. Sipola, “Adaptive framework for network traffic classification using dimensionality reduction and clustering,” *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 274–279, Oct. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6459678

[37] S. Doraisamy and S. R. Uger, “Robust Polyphonic Music Retrieval with N - grams,” *Journal of Intelligent Information Systems*, vol. 21, no. 1, pp. 53–70, 2003.

[38] A. L. Uitdenbogerd, “N-GRAM PATTERN MATCHING AND DYNAMIC PROGRAMMING FOR SYMBOLIC MELODY SEARCH,” in *Proceedings of the Third Annual Music Information Retrieval Evaluation eXchange*, 2007.

[39] S. Buthpitiya, Y. Zhang, A. K. Dey, and M. Griss, “n-gram Geo-Trace Modeling,” in *Pervasive'11 Proceedings of the 9th international conference on Pervasive computing*, 2011, pp. 97–114.

[40] G. K. Zipf, *Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology.* Oxford, England: Addison-Wesley Press, 1949.

[41] C. Tullo and J. R. Hurford, “Modelling Zipfian Distributions in Language,” in *Language Evolution and Computation Workshop/Course at ESSLLI*, S. Kirby, Ed., 2003, pp. 62–75.

[42] C. H. Wu, S. Zhao, H.-l. Chen, C.-j. Lo, and J. Mclarty, “Motif identification neural design for rapid and sensitive protein family search,” *Bioinformatics*, vol. 12, no. 2, pp. 109–118, 1996.

[43] J. Buhler and M. Tompa, “Finding motifs using random projections.” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 9, no. 2, pp. 225–42, Jan. 2002. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12015879

[44] B. Chiu, E. Keogh, and S. Lonardi, “Probabilistic discovery of time series motifs,” *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, p. 493, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=956750.956808

[45] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, “Towards parameter-free data mining,” *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, pp. 206–215, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1014052.1014077

[46] Y. Tanaka, K. Iwamoto, and U. Kuniaki, “Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle,” *Machine Learning*, vol. 58, no. 2-3, pp. 269–300, 2005.

[47] L. Ye and E. Keogh, “Time Series Shapelets : A New Primitive for Data Mining,” in *KDD’09: International Conference on Knowledge Discovery & Data Mining*, 2009, pp. 1–9.

[48] G. Neve and N. Orio, “INDEXING AND RETRIEVAL OF MUSIC DOCUMENTS THROUGH PATTERN ANALYSIS AND DATA FUSION TECHNIQUES,” in *ISMIR 2004: 5th International Conference on Music Information Retrieval*, 2004, pp. 1–6.

[49] J. Lin, E. Keogh, L. Wei, and S. Lonardi, “Experiencing SAX: a novel symbolic representation of time series,” *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, Apr. 2007. [Online]. Available: http://www.springerlink.com/index/10.1007/s10618-007-0064-z

[50] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[51] J. MacQueen, “SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 233, no. 233, 1965, pp. 281–297.

[52] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, “NP-hardness of Euclidean sum-of-squares clustering,” *Machine Learning*, vol. 75, no. 2, pp. 245–248, Jan. 2009. [Online]. Available: http://link.springer.com/10.1007/s10994-009-5103-0

[53] S. Ray and R. H. Turi, “Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation,” in *4th International*

Conference on Advances in Pattern Recognition and Digital Techniques, 1999, pp. 137–143.

[54] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J.R. Statitic. Soc. B*, vol. 63, no. Part 2, pp. 411–423, 2000.

[55] C. A. Sugar and G. M. James, "Finding the number of clusters in a data set : An information theoretic approach," *Journal of the American Statistical Association*, vol. 98, pp. 750–763, 2003.

[56] M. Yan, "Methods of Determining the Number of Clusters in a Data Set and a New Clustering Criterion," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2005.

[57] Q. Zhao, M. Xu, and P. Fränti, "Sum-of-Squares Based Cluster Validity Index and Significance Analysis," in *ICANNGA 2009: 9th International Conference on Adaptive and Natural Computing Algorithms*, 2009, pp. 313–322.

[58] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. MIT Press, 2001, pp. 849–856.

[59] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral Relaxation for K-means Clustering," *Neural Information Processing Systems*, vol. 14, no. Dec, pp. 1057–1064, 2001.

[60] C. Ding and X. He, "K-means Clustering via Principal Component Analysis," in *21st International Conference on Machine Learning*, 2004, pp. 1–8.

[61] I. S. Dhillon, "Kernel k-means, Spectral Clustering and Normalized Cuts," in *KDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 551–556.

[62] I. Dhillon, Y. Guan, and B. Kulis, “A Unified View of Kernel k-means, Spectral Clustering and Graph Cuts,” *UTCS Technical Report #TR-04-25*, vol. June, pp. 1–20, 2004.

[63] “Mobile technologies GSM,” 2001. [Online]. Available: http://www.etsi.org/index.php/technologies-clusters/technologies/mobile/gsm

[64] L. Dryburgh, *Signaling System No. 7 (SS7/C7).* Cisco, 2005.

[65] “Long Term Evolution,” 2005. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/mobile/long-term-evolution

[66] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, *Diameter Base Protocol.* IETF, 2012.

[67] G. Ghidini, S. P. Emmons, F. A. Kamangar, and J. O. Smith, “Advancing M2M communications management: A cloud-based system for cellular traffic analysis,” in *Proc. of the 15th IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM)*, 2014.

[68] R. Coluccio, G. Ghidini, A. Reale, P. Bellavista, S. P. Emmons, D. Levine, and J. O. Smith, “Online stream processing of machine-to-machine communications traffic: A platform comparison,” in *Proc. of the 19th IEEE Symposium on Computers and Communications (ISCC)*, 2014.

[69] “ETSI TS 129 002 V10.2.0,” ETSI, Tech. Rep., 2011.

[70] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” in *20th International Conference on Very Large Data Bases*, Santiago, Chile, 1994, pp. 487–499.

[71] “General Packet Radio Service, GPRS,” 2000. [Online]. Available: http://www.etsi.org/index.php/technologies-clusters/technologies/mobile/gprs

[72] A. Pak and P. Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining,” in *Seventh International Conference on Language Resources and Evaluation*, 2010, pp. 1320–1326.

[73] A. Bifet and R. Gavald, “Detecting Sentiment Change in Twitter Streaming Data,” in *JMLR: 2nd Workshop on Applications of Pattern Analysis*, vol. 17, 2011, pp. 5–11.

[74] R. Batool, A. M. Khattak, J. Maqbool, and S. Lee, “Precise Tweet Classification and Sentiment Analysis,” in *IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, 2013, pp. 1–6.

BIOGRAPHICAL STATEMENT

Stephen P. Emmons received his B.A. in Computer Science from The University of Texas at Austin in 1983. Although he was accepted into the UT Austin PhD program in 1984, he was unfortunately unable to pursue his graduate studies at that time. Instead, he moved to the Dallas area, and later for a few years to the Seattle area, to pursue a successful and rewarding career developing commercial software products, including the award winning Micrografx Designer and Alibre Design. Over the years his area of specialization has ranged across data analysis, business process automation, 2D and 3D computer graphics, image processing, distributed computing, M2M applications, and most recently “Big Data” cloud-based analytics. After putting three children through college – two at The University of Texas at Austin and one unaccountably at Texas A&M University – he chose to pursue his PhD once more.