KU LEUVEN

Faculty of Science
LStat

Predictive Maintenance

In Informationally Sparse Systems

Richard PODKOLINSKI

Supervisor: Prof. dr.Luc De Raedt
Co-supervisor: Wannes Meert

Thesis presented in
fulfillment of the requirements
for the degree of Master of Science
in Statistics

Academic year 2015-2016

"I conceive the mind as a moving thing, and arguments as motive forces driving it in one direction or the other"

John Craig

Acknowledgments

No man is an island.

While a single name appears on the cover of this document, it should not be implied that anything was, or could be done, alone. Everyone is the product of their environments and I've had the esteemed privilege to be assisted and inspired by many wonderful people.

So my great appreciation to Ewa Podkolinska, Jerzy Podkolinski, Alina Kroeker, Raymond Kroeker, Tom Ruette, Wannes Meert, Mathias Verbeke, Gregory Schiano, Jan Aerts, Gerda Claeskens, Luc De Raedt, Roel Braekers, Geert Verbeke, Eric Schmitt, Rui Barbosa, Stijn Van Weezel, Akshat Dwivedi, Irzam Hardiansyah, Zhu Meng, Sytze Elzinga, Justin Fischedick and most importantly Rianne Hartemink.

Thank you for your support and motivation.

Richard Podkolinski
2016
Den Haag
The Netherlands

Intended Audience

Good research is the kind that is capable of taking a complex topic making it accessible to a broader audience. In an effort to adhere to that standard, the intended audience of this thesis is not a pure mathematician, rather a general practitioner in the area of predictive maintenance or reliability analysis. Emphasized is the development of understanding and intuition for the underlying models and how to handle edge cases.

While great effort is undertaken to be as accessible as possible, a certain degree of mathematical development is required to understand this text. Recommended is an understanding of the calculus of probabilities at the level of Blitzstein and Huang[6] as well as a first course on Bayesian linear models at the level of McElreath[48]. A basic understanding of how basic linear models are fit and their performance evaluated is required.

To ensure access to rigorous results, an extensive list of references and resources are made available at the end of each chapter, dubbed "Extended Reading". These resources elucidate many of the technical aspects of the modeling process and mathematical foundations which are beyond the scope of this text.

Abstract

The evolution of predictive maintenance strategies is not universally applicable in all domains. The increased ability of organizations to capture a wider breadth and greater volume of data has lead to a narrow research focus that cannot be easily translated to all domains. Industrial-scale solar power is one of these domains, which has limited access to condition monitoring data. Yet, the need in the domain for predictive maintenance is greater than ever.

This thesis addresses the development of a predictive maintenance process in informationally sparse systems, specifically using the photovoltaic inverters as the object of interest. It seeks to fill the void in the current literature on predictive maintenance in environments where there is limited access to operational data. It addresses the problem through the use of Bayesian time-to-event analysis, a statistical modeling approach used to estimate the remaining lifetime of an object. As such, it progresses through the process of developing a predictive maintenance system starting from the mathematics of time-to-event analysis through to the practical concerns of deploying an process. Finally, it presents an application using simulated data using Stan, the probabilistic programming language.

List of Abbreviations

PV - Photovoltaic

BOS - Balance of Systems

AC - Alternating Current

DC - Direct Current

kWh - KiloWatt Hours

pdf - Probability Density Function

cdf - Cumulative Distribution Function

MLE - Maximum Likelihood Estimation

MCMC - Markov Chain Monte Carlo

HMC - Hamiltonian Monte Carlo

C-Index - Concordance Index

WAIC - Widely Applicable Information Criterion

DIC - Deviance Information Criterion

MSE - Mean-Squared Error

MAE - Mean-Absolute Error

AUC - Area Under Curve

ROC - Receiver Operating Conditions

List of Symbols

T - Random variable for a lifetime

t - Realization of the above random variable; an instance of time.

Y - Random variable for observed lifetimes

y - Realization of the above random variable; an observed instance of time.

d - Censoring Indicator

θ - Vector of Unknown Parameters

θ^s - Vector of Parameters from Completed Simulations

$\Gamma(\cdot)$ - The Gamma Function

$f(\cdot)$ - Probability Density Function

$F(\cdot)$ - Cumulative Distribution Function

$S(\cdot)$ - Survival Function (Reliability Function)

$h(\cdot)$ - Hazard Function (Conditional Failure Rate, Intensity, Force of Mortality)

$H(\cdot)$ - Cumulative Hazard Function

$h_0(\cdot)$ - Baseline Hazard Function

$H_0(\cdot)$ - Baseline Cumulative Hazard Function

$I(\cdot)$ - Indicator Function

$|\cdot|$ - Cardinality of a Set

$g(\cdot)$ - Undefined Lifetime Offset Function

$\mathfrak{g}(\cdot)$ - Undefined Frailty Distribution

$\mathbb{P}_c$ - Probability of Concordance

$\mathbb{P}_d$ - Probability of Discordance

$\mathcal{G}$ - A Graph

$\mathcal{V}$ - Set of Graph Vertices

$\mathcal{E}$ - Set of Graph Edges

elpd - Expected Log Pointwise Predictive Density for a New Dataset

lpd - Log Pointwise Predictive Density

$\hat{p}$ - Effective Number of Parameters

Contents

Introduction

Maintenance is not sexy, but it is essential. It is the immune system of the modern industrialized world. Without it, the myriad of systems we depend on for safeguarding our collectively quality of life would rapidly deteriorate and disappear. Yet, few sing the praises of the multitude of technicians and engineers which are responsible for ensuring trains run on time, computer systems remain accessible or that bridges do not suddenly collapse. Paradoxically, this lack of visibility is a testament to the uninterrupted success of these activities.

The preceding decades have brought with them rapid innovation in the area of maintenance. What has historically been a manual and labor-intensive activity is increasingly becoming a scientific process, one that leverages technology to optimize operations. Modernization has meant going beyond traditional methods of maintenance based on scheduled inspection or reactivity and moving towards those that utilize data and act before failure occurs. Such a trend can have a widespread impact on operational expenses. Reactive methods generally incur greater costs due to downtime and the need for more expensive replacement parts, due to shipping. Scheduled inspections share similar pitfalls, requiring downtime, specialized machinery and also running the risk of inadvertent damage to the objects they seek to maintain. Predictive maintenance minimizes many of these drawbacks. Judgments about the state of a device are made as a result of continuous monitoring of actual operating conditions[50]. This reduces downtime and allows for a better allocation of maintenance resources. Most importantly, predictive maintenance enables faults to be detected *before* failure allowing for contingencies, such as repair or replacement, to ensure constant output.

The evolution of predictive maintenance owes much to the continued development of sensor technology. More reliable, accurate, and most importantly, inexpensive sensors have enabled continuous monitoring in a wider range of domains. Many modern devices come with built-in sensors that record output or other context-specific data that can be used to assess health. Increasingly, many also include connectivity that allows telemetry to be retrieved remotely and on demand. This reality is likely what has spurred widespread innovation in the field, with more than half of the publications on predictive maintenance emerging within the last decade.[1] Unfortunately, this data gluttony has also led to a form of tunnel vision, with the majority of this work dedicated to enabling predictive maintenance exclusively in informationally rich contexts. As a result, authors are predominantly preoccupied with domains where data is high dimensional, time-dependent, with fine granularity and encoded without error.

While the future may make these ideal circumstances ubiquitous, there is still a wide set of domains where this is not a reality. Sensors may have become increasingly cheap

[1]Based on Search: Limo = 703 / 989 and Google Scholar = 3300 / 5450

and plentiful but they can still be expensive in relative terms. The majority of the sensors included in new products are meant to extend functionality and the continuous monitoring they enable is an fortunate but inadvertent consequence. Yet, for many devices the inclusion of additional sensors means an increase in manufacturing costs, which is often prohibitive. No product developer of sound mind would argue for the inclusion of a sensors in a t-shirt to prevent stitching failure. The sensors would vastly increase the final cost of the original product. However, this kind of cost-benefit dilemma is found throughout numerous domains where the sensor costs are high relative to the manufacturing cost of the product. In these domains, data is rarely high dimensional, time-dependent, fine and encoded without error. Despite these limitations, there is still a need for predictive maintenance.

This thesis addresses the development of predictive maintenance processes in informationally sparse systems, specifically using photovoltaic inverters as the object of interest. In so doing, it will attempt to fill the void in the current literature on predictive maintenance in domains where limited access to operational data is the norm. It will address the problem from the perspective of Bayesian time-to-event analysis, a statistical modeling approach which is used to estimate the remaining lifetime of an object. It will develop the methodology from the ground up, starting with mathematical foundations and leading to practical concerns in deployment of such an process. Finally, it will provide an application using simulated data and Stan, the probabilistic programming language. The following chapters encompass the development of this strategy.

The first chapter examines predictive maintenance and introduces an example domain meant to provide a business case. The solar industry with its heavy reliance on dwindling government subsidies and firm need to reduce costs proves an ideal domain for such processes. Further, the photovoltaic inverter, a device with limited output found in every solar power plant is a highly relevant object of study.

The second chapter considers methodology, providing the probabilistic and statistical foundations required to understand Bayesian time-to-event models. These foundations are then used to introduce more complex models centered around the Multiplicative Hazard Model with shared Frailties. Finally, estimation procedures for these models using Bayesian Methods are introduced with a brief description of Hamiltonian Monte Carlo.

The third chapter covers the analytical process using simulated photovoltaic inverter data. It restates the goal of the analysis in concrete terms, then examines the steps needed to develop a predictive maintenance process. Feature engineering is examined, including what resources must be made available for such a system to be put into place. Then, data management and the simulation process are discussed. Stan, a recently development in fitting Bayesian models is examined and its implementation method is described. Then, an exploration of how to assess the predictive performance in the context of time-to-event models is provided. The chapter concludes with a step-by-step construction of the time-to-event models in Stan. Finally, a summary of the material is provided.

Domain

It is impossible to perform an analysis without a thorough understanding of the context in which it is performed. At its heart, the purpose of statistics and machine learning is to glue the crystalline dimension of mathematics to a messy and unstructured world. A comprehensive understanding of the conditions of that world are required for this purpose to be successful. This ensures that the models built accurately represent what they seek to influence. Models for predictive maintenance are no exception.

This chapter covers the domain in which a predictive maintenance process is developed. It begins with an introduction to the solar industry, its role and current state and its need for further cost reduction. It then turns its attention to predictive maintenance and provides the aims and requirements for the process. Next, the photovoltaic system is explored. Its composition, data sources and challenges in enabling predictive maintenance are discussed. Finally, a brief overview of the simulated data and model justification are provided.

Solar Industry

Altering the global energy mix is essential to the future evolution of civilization and possibly the survival of the species. Currently, the global energy mix is dominated by fossil fuels. Approximately, 82% of global energy consumption remains the product of coal, natural gas and petroleum[17]. A century of heavy reliance on these non-renewable fossil fuels as the central source of energy has had severe ecological and socio-economic consequences. Locally, there have been countless instances of environmental damage, such as air[69] and groundwater[62] contamination and oil spills[78]. Additionally, regions rich in fossil fuels are destabilized as wealthy nations seek to guarantee future supplies[64]. Globally, the prospect of catastrophic climate change looms large, with implications including rising sea levels[14], ocean acidification[19], crop failure[55] and the economic devastation of a bursting carbon bubble[9].

Despite these persistent consequences, a reduction in consumption is highly unlikely. Global energy demand is projected to increase rapidly in the coming decades. As of 2012, global demand was estimated at $5.8 \cdot 10^{20}$ Joules and is expected to rise by 48% to $8.6 \cdot 10^{20}$ by 2040[17]. The majority of this increase is the product of higher demand for electricity, which will increase by 70% before 2040. The majority of this new demand, roughly 87%, is expected to come from countries in the developing world[54]. Due to their economic conditions, these countries will undoubtedly utilize whichever energy source is cheapest, regardless of the consequences.

In this context, inexpensive alternative energy production, sourced from renewable

resources, is required to evade a future of socio-economic instability and ecological collapse. Fortunately, an increasing number of alternatives are being developed with the support of governments. Solar, wind, tidal, geothermal and fusion have all seen major investments over the past decade. In 2015 alone, renewable power capacity received twice the investment compared to new projects based on fossil fuels[22]. Of these projects, wind and solar appear the most likely to become viable within the next decade[53].

The contribution of solar energy is particularly attractive as its supplies are not limited by resource availability. At least for the next five billion years, solar energy is expected to reach the earth without interruption. Furthermore, the amount of solar radiation intercepted by the earth is several orders of magnitude higher than global energy consumption[25]. Given such potential, it is not surprising that the rate of installations of solar systems has increased exponentially in the past decade[47].

The central drivers of this growth have been continued technological development and innovative government policy. In terms of technological development, three major factors contribute to growth. First, the continued improvement of efficiency of end-use technologies. This ensures that electrical output is utilized optimally by the time it reach consumers[79]. Second, the reduction in cost and enhancement of energy storage technologies[34]. This ability is particularly important for the solar energy industry as production fluctuates based upon the availability of sunlight. Finally, and most importantly, the reduction of costs associated with technologies that convert solar energy into electricity.

In terms of reducing costs and optimizing production, the advances continue. There are frequent discoveries of novel materials which enable broader capture of solar radiation[3]. As well as developments that optimize power processing systems, ensuring that energy produced can be efficiently transfered back to the power grid[76]. These technological improvements have led to a substantial decrease in the costs associated with photovoltaic power. The average weighted utility-scale solar photovoltaic installation cost has more than halved in the last five years[35].

However, further cost reductions are needed. The central bottleneck for greater expansion of solar power continues to be economic self-sustainability. In most markets, solar is not yet able to compete with other forms of energy generation without government policy incentives[52]. These usually come in the form of subsidies or feed-in tariffs that provide solar companies with long-term contracts for energy above market value. These are meant to provide stability as well as offset the cost of further technological development[13]. However, these subsidies are reduced incrementally by design, as they are meant to serve as a temporary incentive to promote cost-saving innovation. As the subsidies are reduced, the industry is more subject to market forces and more reliant on its own ability to optimize its efficiency. The current low price of oil and coal, coupled with fickle government support for renewable energy[22] present a difficult status quo[28].

While technological innovation is likely to continue to reduce costs, there are avenues for further optimization given the current infrastructure. Among them are improvements in reliability, specifically in photovoltaic systems. Reliability has been a long-standing challenge[57] and a particularly important one given systems long payback periods[51]. Innovation in components and manufacturing are methods for addressing reliability, but other more immediate options exist. Among them, optimized maintenance activities have the potential to dramatically reduce costs. Yet, these optimizations require predictions about the state of component health. Therefore, predictive maintenance is required.

Predictive Maintenance

Predictive maintenance is generally defined as the process of using direct monitoring of conditions, efficiency and other indicators to determine the health and likelihood of failure of a system. These results indicate when maintenance tasks should be performed[50]. Informally, predictive maintenance is an activity geared toward indicating the degree of wear of a piece of equipment and predicting its useful life[42]. In doing so, it empowers organizations with the ability to plan and act against operational hazards. This additional time allows for corrective actions to be prepared including the ordering of parts and materials, often at lower prices, coordinating labor and scheduling downtime. In turn, this allows the business, not the environment, to dictate when service interruptions occur.

All predictive maintenance processes attempt to address one or more of the following three generic use cases[72]. First, is fault identification. Accurately identifying the source of failure of a particular system is essential to developing effective corrective countermeasures. Second, is near-term failure probability. This permits immediate operational risks to be addressed and avoided. Third, and finally, is estimation of residual lifetime. This allows both short and long-term operational risks to be taken into account. All the cases overlap and should not be treated as mutually exclusive. Yet, each requires slightly different considerations, methodology and inputs when developing a predictive maintenance process.

The preconditions for the development of a predictive maintenance process differ depending on the domain and selected use cases. Yet, there is a common core of needs including the ability to implement results, high data quality and a typical set of maintenance related sources.

The ability implement a corrective mode of action is universal requirement. Without the ability to act, no amount of information derived from a predictive model will be of any value. After all, predictive maintenance does not, in and of itself, improve the reliability of operational systems, only maintenance activities do[42].

As predictive maintenance depends on condition monitoring, data sources of a suitable quality are required. Ideally, data sources should be historical, abundant, balanced, relevant and noiseless. Models that generate predictions about the future require historical data from which to derive patterns about operating conditions. The more historical information is recorded, the greater the opportunity of the model to observe the system along its lifetime. This in turn results in more accurate predictions about future health. Furthermore, the more abundant the data sources the better. As a greater number of repeated observations provides a greater certainty about the state of a system at any given time. Yet, in this context, historicity is of greater importance than abundance. It is possible to be tracking a large quantity of systems, but only for a short period of time. This would be inadequate due to a natural imbalance in the data for this type of process. Even in systems with profoundly poor reliability, healthy observations will greatly outnumber failures. Without a greater quantity of historical data, failures are even less likely to be present in the data. Therefore, a degree of balance is required in the data between the number of observed failures and healthy observations. Additionally, data sources must be relevant to the task. Data sources which offer the greatest potential for predictive power should be utilized. Domain knowledge from subject matter experts should be exploited to provide a contextually appropriate selection. These experts should

also implement mechanisms that ensure data sources are as free as possible from measurement error, missing values and other forms of noise. While unavoidable in a practical setting, noise can undercut the reliability of models and introduce a greater degree of uncertainty in predictions.

A typical set of data sources for a predictive maintenance process includes system attributes and conditions, maintenance logs and failure history[72].

Attributes are a set of static values that define the identity of the system. This includes elements such as a system's ID, model number, date of installation, location, a list of components as well as any other technical specifications that might be relevant to its reliability. As these attributes are unlikely to change over time, they provide context which allows for the comparison of systems with one another.

Conversely, conditions are temporal values that define the state of a system at specific points in time. These values are captured through telemetry from sensors or other periodic observations. They can include, but are not limited to, component status, yields, environmental conditions or any other time-varying data directly related to reliability. These conditions are meant to capture the evolution of a system as it ages and provide the degradation pattern leading up to failure.

Maintenance history logs contain all service interactions with any particular system. These time stamped logs should contain any inspections, identify suspicious or replaced components and provide a listing of any corrective activities that maintenance teams performed. This data is crucial in providing context for the health of any system over time. It not only allows for the determination of the efficacy of maintenance tasks but also improves the ability to better understand the sequence of events leading up to failure.

Finally, failure history includes the time and conditions of observed breakdowns across systems. These values are usually entered manually upon system failure. An accurate and timely declaration that a machine has failed and the cause of that failure is central to the task of predictive maintenance. A generic recording of failure, without determining the cause, is not sufficient. Utmost care must be taken to ensure that the failure history records are current and complete. Delays in recording failure times detrimentally impact predictive performance. This is likely the single most important data source for a predictive maintenance process. If recorded inaccurately, it will undermine the entire endeavor.

Predictive Maintenance in Photovoltaic Inverters

Predictive maintenance presents an opportunity for improved cost optimization in the solar industry. This section examines one area of application within the industry, namely predictive maintenance of photovoltaic inverters. First, a general picture of a typical photovoltaic system at the utility-scale is given, demonstrating the role of inverter reliability in its operation. Second, a description of the considerations when developing a predictive maintenance process in this domain are given.

As can be seen in Figure 1, a typical photovoltaic system is composed of two main elements, one is a solar array and the other are balance-of-system components. A solar array is usually composed of individual photovoltaic cells wired in series, called a string. This is done to increase the voltage produced by individual cells. The strings are then wired in parallel to increase the current. This collection of strings is then encapsulated

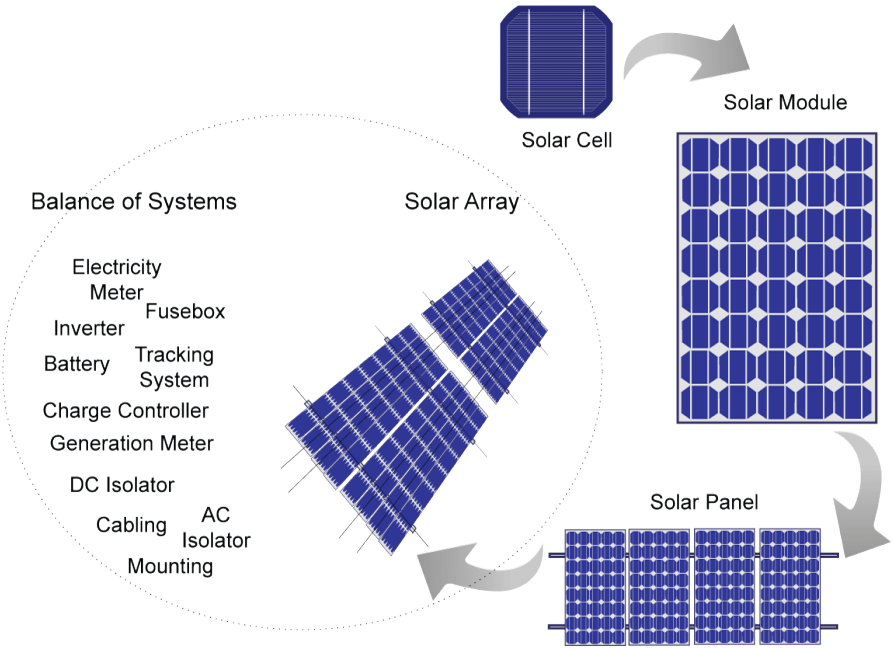

Figure 1: Photovoltaic System

in a weather-resistant housing and is what we typically refer to as a solar panel. These panels are then connected to one another to form a solar array. The connection can be done in either series or in parallel depending on the desired voltage or current of the system.

Balance-of-system (BOS) components encompass everything that is not the solar panels or array. The BOS components include the array structure, solar tracker, connectors, AC and DC wiring, over-current protections, disconnects, interconnects, charge and system controllers, maximum power point trackers, batteries, inverters and any other accessories. The purpose of the BOS components is to integrate the solar system into the utility grid.

A photovoltaic inverter is an electrical device that converts the variable DC output of the solar panels to an AC frequency allowing it to be fed to the utility grid. Of all the BOS components, the inverter has historically had the most reliability problems. A international decade-long survey of photovoltaic systems indicated that a majority of failures, 65%, were attributed to inverters[41]. The same research also stated there appears to be ongoing reliability improvements, as the rate of failure declined over time. Even with these improvements, recent research confirmed that inverters continued to be the most likely causes of photovoltaic system failure[26].

Inverter reliability is essential for ensuring steady output of a solar system. When an inverter fails, it disables all photovoltaic cells upstream of it. This makes it the single largest potential source of productivity loss in a solar system. It is for this reason, a predictive maintenance process focused on the photovoltaic inverter can serve as a valuable mechanism for optimizing operational outputs.

Yet, there are a number of difficulties in deploying such a process within this domain. Given its role as the device connecting the solar panels to the utility grid, the inverter

contains most of the intelligence in a photovoltaic system[26]. However, this intelligence is minor and narrowly focused. At any given moment, an inverter records its electrical output in kilowatt hours (kWh) and possibly provides an error code if a malfunction has been detected[63]. Unfortunately, these error codes are noisy and are not always present prior to an inverter failure. Even when an error code is present, it does not uniquely define a specific cause of failure. It, instead, expresses a symptom of failure, such as current leakage, over-voltage or software error, rather than the underlying cause. Furthermore, there is a tacit acknowledgment of the lack of causality between the presence of these error codes and inverter failure. This can be seen in most dashboards of solar systems which will report a count of the number of error codes that have occurred throughout the day. Thus, a single error is not seen as particularly severe and, as stated earlier, may not even occur prior to inverter failure.

Additionally, communication errors are relatively common with system condition data, at a rate even higher than inverter failure[26]. This results in missing values being present in historical data. As these missing values are of electrical output and error logging, it results in confusion as to whether or not the system has experienced failure or not. It also makes it impractical to use interruptions in system conditioning data as a means of detecting failure. It is an obligation of system operators to provide logs for such downtime so that this distinction can be made. However, given their frequency, and their requirement for human intervention, this often does not occur. Yet, improvements of system network connectivity are likely to resolve this problem in the near future, both in terms of on-site data collection and continuous transmission.

The lack of robust system condition data poses a challenge in developing a predictive maintenance process. At its core, the lack of condition data stems from the modularity of design in photovoltaic systems. Every additional component not directly related to generating electrical output is viewed as an operational overhead, requiring cost-benefit justifications which often do not fit into profitability planning. As a result, the inclusion of accurate sensors which track reliability indicators like temperature and humidity at the inverter level are viewed as cost prohibitive additions and generally not included. This leads to informational sparsity in photovoltaic systems and especially in inverters.

This can be contrasted with another alternative energy source, wind, which by design does not face such problems. A wind-turbine has roughly a hundred sensors that measure everything including ambient humidity, temperature, airflow and electrical current[10]. Of course, wind-turbines are self-contained and far less modular compared to a photovoltaic system. However, other industries, such as cloud computing, that have similar modular designs do not share the same sensor deficiencies. For example, the average cloud storage server rack contains more than a hundred temperature sensors, with each server averaging three[56]. The key is that the monitoring requirement of a cloud computing server rack and a solar system are comparable. Yet, there is a substantial difference in the application of sensors to create a cohesive picture of health for each unit. The cloud storage industry clearly believing that system conditioning data is worth the overhead expense.

As the industry matures and solar power becomes independently profitable, it is likely that it too will invest more heavily in condition-based monitoring of photovoltaic inverters[20]. However, this is not yet the case. Most sensors in a solar system are at the scale of solar park as a whole, not at the scale of an inverter. Thus, temperature, humidity and solar radiance data, assuming they are recorded at all, are provided by park level sensors. With large solar parks having multiple sensors installed. This is remark-

ably suboptimal, as solar parks at a utility-scale can encompass a large landmass, some exceeding a hundred square kilometers. A dozen sensors scattered among that landmass will undoubtedly produce a noisy signal regarding the health of any individual inverter. There are also attempts to reduce costs of sensors by employing satellite sensor data. However, this provides even worse resolution than park-level sensor data and thus even noisier signals regarding the health of an inverter.

While the system itself may not record a substantial amount of useful data, the humans working with the systems often do. This provides contextual information which can be used in the construction of a predictive maintenance process.

System attributes are almost always kept complete and up to date. The make, model, installation date, location and other static data are recorded. Utility-scale solar companies often have warranty contracts with manufacturers which depends on the accuracy of this data[27]. This often includes offers to replace faulty elements within a specific time of purchase.

Maintenance logs and failure history are usually kept but the level of detail depends on an organization's commitment and awareness of the utility of this data. Some companies chose not to employ maintenance logs at all, instead opting for run-to-failure operations. As the name suggests, components in the system are run until they fail and are then replaced. This is an suboptimal reactive approach, and also the most expensive form of maintenance. It leads to to increased downtime and higher costs of repair parts[50]. Companies that run-to-failure will usually still store accurate failure times for these components but not always. This can result in discrepancies between failure times and a lack of system electrical output leading to confusion about exact timing of any failure[41]. Companies that invest in some level of preventative maintenance will keep logs which include what components were inspected and what, if any, corrective measures were taken. However, the level of detail of these logs varies highly. If kept, some degree of high-level root cause is almost always included. This may be at a level that is not sufficiently discriminating such as the use of "parts and material" or "external cause" as failure categories. Some companies, especially the large ones, do provide further context for these failures, but they are usually still fairly high level, such as "control software" and "AC contactor". However, limited willingness to engage in deep-failure analysis can dampen the value of collected information[26].

As is clear, there are some significant hurdles to overcome in the development of a predictive maintenance process within the solar energy domain. However, as the industry matures and becomes increasingly self-sustainable it will undoubtedly correct many of these imperfections.

Simulated Data and Model Justification

The purpose of this section is to broadly lay out the simulated data set used throughout the remainder of the text and justify the use of a Bayesian time-to-event model.

The simulated data uses the basic designs of a real-world photovoltaic system, but lacks the faults generated by poor data management, connectivity issues and other administrative errors. The data sources of several photovoltaic companies were used as the basis for the simulated data. For the sake of exposition, the data is a simplification of reality, in that it does not cover many of the possible externalities that arise in such a system. The purpose, after all, is to demonstrate how a predictive maintenance process can be developed rather than focusing on eliminating edge-cases. Furthermore, the use of simulated data provides the ability to iteratively construct a usable model and add complexity as required.

It contains enough similar features to allow for it to be considered a reasonably representative data set. Five-hundred inverters unevenly clustered in five solar parks are generated. Each of the five-hundred inverters is provided with an ID, a model type, and a park. One generic failure mode is used, which occurs within the first year of operation, representing sources of infant mortality commonly found in inverters[16]. The same type of process could just as easily applied to old age wear or any other failure mode. However, given that inverters expected lifetime is a decade, this would insert added complications related to technological innovation in that period which are beyond the scope of this text.

Electrical output data is provided, with both trend and seasonality removed. Furthermore, error codes are provided. Two generic error codes are generated whose combinations are related to the aforementioned failure mode. Park level data is also provided. Temperature and humidity measurements are provided at the solar park level, which are combined into an extreme weather event count.

Finally, maintenance logs are compressed into a single activity, related to component repair or replacement within a photovoltaic inverter. It should be stressed that replacement refers to subcomponents of the inverter, like capacitors, not the inverter itself. A count of the days since the last instance of such an activity is given.

From this description, it is clear that this simulated data could be infinitely more complex. In a real-world setting, there are a greater number of failure modes, error codes and maintenance activities. However, in order to derive the essence of predictive maintenance process, this level of simplification is justified. Additionally, a real-world context is unlikely to have a greater quantity of machine condition data, only further contextual information. Thus, this additional context simply increase the dimensions of the input but does not alter the heuristic required to produce a predictive maintenance process.

There are numerous available techniques for generating predictions of remaining lifetime. However, most of these techniques require a greater volume of condition monitoring data than is available in an average photovoltaic system. The selection of Bayesian time-to-event analysis is based on a desire to optimally use the information that is available.

The use of time-to-event models is an acknowledgment that any system's history informs the state of any other system at similar stages of deterioration. While other methods exist that could be used for the development of a predictive maintenance process, like multiclass classification models or multiclass logit models or regression trees, none of these

methods operate well in a context with a limited amount of censored data. Yet, Bayesian methods make optimal use of this small amount of available data and can be adapted to deal with censoring. Furthermore, these methods provide valuable insight into the nature and composition of breakdowns. Thus, they provide valuable information that can be used for corrective and preventive purposes outside of the predictive process.

Extended Reading

For an understanding of international energy policy and the role that solar power plays, the OECD's World Energy Outlook is a good place to start[17]. For an in-depth, but dated, introduction to predictive maintenance, the reader is directed to Mobley[50]. For a general view of constructing predictive maintenance solutions in informationally rich domains, Microsoft has assembled an excellent step-by-step guide[72]. For an overview of the reliability issues specifically related to photovoltaic systems, the reader is directed to Golnas[26] and Petrone[57].

Methodology

Time-to-event analysis[2] concerns itself with the modeling of the expected duration of time until an object experiences a well-defined event. This event is a transition between a finite number of possible states, such as from operational to non-operational or from alive to dead[2]. In this particular context, these states define the transition of a photovoltaic inverter from functional to failed.

To be able to model the time until these transitions, precision in describing phenomenon is needed, and for that, mathematics is required. The following chapter describes the mathematical narrative that underlies the construction, manipulation and estimation of Bayesian time-to-event models. It begins with the basic functions required to understand this class of models, their relationship, and role in defining time-to-event distributions. Then, censoring and truncation which are common characteristics of time-to-event data are examined. From these basics units, the Multiplicative Hazard Model for time-to-event analysis is formulated. Following this, a hierarchical model which describes shared risk, or Frailty is introduced. Finally, a description of the likelihood and model estimation with Hamiltonian Monte Carlo is given.

Time-to-Event Functions

The purpose of this section is to firmly root time-to-event analysis in the broader context of statistical modeling. It demonstrates the mathematical functions required to interact with this class of models, as well as providing their relationships. As this section is foundational, its content can be attributed to a diverse number of sources[1][68][38][39][11][60].

Begin, by defining T, a continuous non-negative random variable with an unknown distribution representing the time until a well-defined event. As it is a time, its support is constrained to all positive real numbers, ($T \geq 0, T \in \mathbb{R}$). Further, define t, to be the realization of this random variable at a specific point in time.

A familiar way of describing a probability distribution is to use probability density function (pdf) and cumulative distribution function (cdf). The pdf is the relative likelihood of a random variable taking a particular realization.

$$f(t) = \Pr(T = t)$$

The cdf defines the probability that a random variable will take on a value less than or equal to some realization. Thus, it defines a range of outcomes across the random variable.

[2]also known as survival analysis, reliability analysis and event history analysis

$$F(t) = \Pr(T \leq t) = \int_0^t f(x)dx$$

In the time-to-event context, the cdf of T may not be particularly useful as the variables of interest usually take on values greater than some realization. This is because in most cases, T will be greater than the value observed, t. Fortunately, by definition, a random variable must sum to one and the complement of the cdf can be established through subtraction.

$$S(t) = P(T \geq t) = 1 - F(t) = \int_t^\infty f(x)dx$$

This returns the survival function[3] which is the probability that the well-defined event occurs after a specific point in time. It is considered the survival function because it provides the probability of surviving beyond an observed time, t. Clearly, if the event occurs after a specific time, then the event has not yet occurred. As the event is death or failure, it is implied that the object of interest has survived up to that point.

As the survival function is the complement of the cdf, it inherits its properties. It is monotonically decreasing, with $S(0) = 1$ and $S(\infty) = \lim_{t\to\infty} S(t) = 0$. This formalizes the notion that when the event in question is failure, in the beginning all systems are operational, but given a long enough time frame all systems will eventually fail.

While the survival function focuses on the event not occurring, the hazard function focuses on the event occurring. The hazard function[4] is the instantaneous rate of failure given that failure has not yet occurred.

$$h(t) = \lim_{\Delta t\to 0^+} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} = \frac{f(t)}{S(t)}$$

It provides the potential of the event occurring within the next limiting time interval, assuming it has not taken place until now. It should be stressed, that this function does not return a probability, rather a rate ($\frac{P}{\Delta t}$). It must be non-negative, such that $h(t) \geq 0$, and can take on values greater than one $[0, \infty)$.

The hazard is far more useful from a practical perspective than the other constructions. The conditional formulation is especially important, as the hazard defines the risk only after excluding the prior occurrence of the event. This makes it a more natural expression of what is generally asserted as the risk of an event in time, which implicitly presupposes that the event has not yet occurred.

Numerically, the hazard is clearly linked to the future occurrence of the event. When the hazard is zero, the risk of the event occurring in the next moment is also zero. Conversely, when the hazard is infinite, the risk of the event occurring the next instance is near certain.

The hazard is a limiting rate, and is concerned with the event within an instantaneous interval. At times, it is beneficial to understand that potential across an interval of time.

$$H(t) = \int_0^t h(u)du$$

[3] also denoted as the reliability function $R(t)$

[4] also known as the conditional failure rate, intensity and force of mortality

This is the cumulative hazard function. It measures the total amount of risk that has been accumulated up to time t. The cumulative hazard can be understood as the number of times we would expect to observe the event in a given period of time, assuming the event were repeatable.

The relationship between the hazard and cumulative hazard is especially important in providing intuition. The hazard is a rate is defined in $\frac{1}{t}$ units. Whereas the cumulative hazard sums across those $\frac{1}{t}$ units. For example, if a rate of a particular event was ten, and five units of time passed, then fifty events would be expected within those five units. It is important to note, that this logic is one-directional and is based on an assumption of a constant hazard rate over the five units of time. There is nothing that makes this generally true. Thus, when starting with a cumulative hazard of fifty, it is possible that the hazard rate is fifteen, five, ten, five, and fifteen, respectively, for each of the five units of time.

The pdf, cdf, survival, hazard and cumulative hazard functions all uniquely define the process generating time-to-event data. As a result, it is possible to transform any of these functions into any other. This is useful because it provides insight into the relationships among the functions but also because it allows for the use of the easiest constructions in modeling.

It is generally true that any pdf can be expressed as a derivative of its cdf. As a result, the survival function also provides a route back to the pdf. The negative derivative of the survival function returns the pdf. Such that the pdf can be redefined as follows:

$$f(t) = \lim_{\Delta t \to 0^+} \frac{\Pr(t \leq T < t + \Delta t)}{\Delta t} = \frac{dF(t)}{dt} = -\frac{dS(t)}{dt}$$

The hazard function and survival function are also intimately related to one another. From the above identity, it can be seen that the pdf in the definition of the hazard can be replaced with the negative derivative of the survival function. The same can be done in the definition of the hazard. It is then clear that the hazard can be expressed as the negative derivative of the logarithm of the survival function.

$$h(t) = \frac{f(t)}{S(t)} = -\frac{dS(t)/dt}{S(t)} = -\frac{d \ln\left(S(t)\right)}{dt} = -\frac{d}{dt} \ln S(t)$$

The above identity can also be used to express the cumulative hazard. Integrating from a starting time until a specific time, demonstrates that the cumulative hazard can be defined as the negative logarithm of the survival function.

$$H(t) = \int_0^t \frac{f(u)}{S(u)} du = -\int_0^t \frac{1}{S(u)} \left(\frac{d}{du} S(u)\right) du = -\ln S(t)$$

Whatever transformation is expressed in terms of logarithms, can be reversed through exponents. Thus, the survival function can be defined as the exponent of the negative integral of the hazard function, or the negative cumulative hazard.

$$S(t) = \exp\left(-\int_0^t h(u) du\right) = \exp(-H(t))$$

This allows for the conversion of a cumulative hazard back into a probability. Given the earlier example of a cumulative hazard of fifty the survival probability can be determined: $S(t) = \exp(-50) = 0.19 \cdot 10^{-22}$ which is a near zero probability of survival.

The pdf can also be expressed in terms of the exponent of the negative integral of the hazard function, or the cumulative hazard.

$$f(t) = h(t) \exp\left(-\int_0^t h(u)du\right) = h(t)\exp(-H(t))$$

As can be seen, regardless which of the functions are available, it is possible to produce a transformation which will return all the others. This is useful because generally, the hazard or survival time rather than the density are the objects of interest.

All of these functions can be conditioned to only extend to events that occur after a particular point in time. This is useful when dealing with lifetimes where the system has not been observed from the time it became operational. It is also useful when constructing expressions about particular intervals of time[11].

$$h(t|T > t_0) = h(t)$$

$$H(t|T > t_0) = H(t) - H(t_0)$$

$$F(t|T > t_0) = \frac{F(t) - F(t_0)}{S(t_0)}$$

$$f(t|T > t_0) = \frac{f(t)}{S(t_0)}$$

$$S(t|T > t_0) = \frac{S(t)}{S(t_0)}$$

One final construction of interest is the expected residual lifetime. This is the amount of life a particular system is expected to have given it has survived up to a specific time (t_0).

$$E(t|T > t_0) = \int_{t_0}^{\infty} S(t)dt \cdot \frac{1}{S(t_0)}$$

Common Lifetime Distributions

Any continuous distribution defined over the positive numbers can be used as a lifetime distribution. This section briefly covers several commonly used distributions for modeling lifetimes and provides justifications for their use. The exponential, Weibull, Gompertz and Gamma are described.

The simplest lifetime distribution can be defined by assuming a constant hazard rate (λ) over time.

$$h(t) = \lambda \qquad (\lambda > 0)$$

As was demonstrated earlier, this can be transformed into a survival function by exponentiating the integral of all values from start until t. As the hazard is constant, this will simply be the exponent of the constant multiplied by the number of intervals present.

$$S(t) = \exp\left(-\int_0^t \lambda \, du\right) = \exp(-\lambda \, t)$$

If transformed into a density, its characterization becomes immediately apparent.

$$f(t) = -\frac{dS(t)}{dt} = \lambda \, \exp(-\lambda \, t)$$

This is the functional form of exponential distribution. It is useful in two areas in time-to-event analysis. First, it can be used to model events whose likelihood of occurrence does not vary with time. It can model systems that are not affected by wear or aging. For example, the exponential distribution is a suitable model for the release of particles from radioactive material[36]. However, these types of events are relatively rare in industrial settings. There are very few systems that do not experience some changes as a result of their length of operation. The second area of use for the exponential distribution is the discretizing of time. Time is continuous but data is not. Thus, there is a need to define how continuous time behaves across recorded intervals. The most consistently used assumption is that the hazard does not change within any given interval. If these intervals are sufficiently small such an assumption is perfectly valid.

Most lifetime distributions arise as generalizations of the exponential. They provide a more elaborate construction for the hazard over time based on some underlying logic.

The Gompertz distribution is a generalization of the exponential that introduces an exponential effect in the hazard over time.

$$h(t) = \lambda \exp(\varphi t) \qquad (\lambda > 0, \varphi \in (-\infty, \infty))$$

The formula was originally derived to characterize the exponential rise in death rates in humans between sexual maturity and old age[77]. It introduces a shape parameter φ which controls the rate of change of the hazard over time. When $\varphi < 0$ the hazard declines over time. This can be used to model component "burn-in", which is the time shortly after beginning operations where mechanical parts often experience early failures. Conversely, when $\varphi > 0$ the hazard increases over time. This can be used to model end of life failure. If $\varphi = 0$, the Gompertz reduces to an exponential distribution. When the Gompertz hazard is not constant, it is always increasing or decreasing over time. This makes it particularly attractive when it is clear that operational time is the greatest force of mortality.

The Weibull distribution also introduces an exponential effect in the hazard function over time. Yet, it does so in a more flexible manner.

$$h(t) = \lambda \nu t^{\nu - 1} \qquad (\lambda > 0, \nu > 0)$$

It introduces a shape parameter ν that also controls the increase or decrease of the hazard over time. It is more flexible than the Gompertz in that it is capable of modeling hazards that increase initially, but whose rate of increase declines over time when ($0 < \nu < 1$). The Weibull also has a common alternative parameterization, which is common in many software packages.

$$h(t) = \frac{\alpha}{\sigma}\left(\frac{t}{\sigma}\right)^{\alpha-1} \qquad (\alpha > 0, \sigma > 0)$$

This parameterization defines a shape, α, and a scale, σ, parameter. These parameters are equivalent to the rate, λ and shape, ν, in the earlier form such that, $\lambda = \sigma^{-\alpha}$ and $\nu = \alpha$.

The Weibull distribution was originally developed to assess the catastrophic failure of materials. No material is perfectly uniform and all contain irregularities at some scale. These irregularities such things as pores, mineral inclusions or micro-cracks are distributed throughout the material. While under pressure any one of these irregularities can induce failure. As a result, the Weibull formalizes the notion of the "weakest link"[59]. It is the minimum of any collection of independent identically distributed random variables. This is important in the reliability of complex systems as different causes of system failure compete with one another and the first cause will result in the failure of the entire system[58].

Gamma distribution, like those before it, is also a generalization of the exponential. However, its justification for use as a lifetime distribution is a bit more involved. A system may be exposed to a number of shocks, each of which are exponentially distributed. The system may be resilient to each shock up to a threshold, upon which it fails. The sum of each of those exponentially distributed shocks is gamma distributed[71]. The hazard function is as follows:

$$h(t) = \frac{\lambda^k t^{k-1}\exp(-\lambda t)}{(1 - I_k(\lambda t))\Gamma(k)} \qquad (k > 0, \lambda > 0)$$

With k being the number of exponentially distributed shocks that occur prior to system failure. As can be seen the hazard is quite complex, making use of both the gamma function $\Gamma(k)$ and the incomplete gamma integral $I_k(\lambda t)$.

$$\Gamma(k) = \int_0^\infty x^{k-1}\exp(-x)dx \qquad I_k(x) = \frac{\int_0^x s^{k-1}\exp(-s)ds}{\Gamma(k)}$$

In traditional model fitting techniques, the incomplete gamma function imposes numerical problems for parameter estimation[77]. This has resulted in the gamma function not finding widespread application in the modeling of lifetimes. Recently there has been a resurgence in the use of the Gamma distribution, especially in the Bayesian context where optimization problems are less of a hurdle to application. Additionally, hierarchical modeling makes use of the Gamma distribution as method of modeling unobserved heterogeneity among groups of lifetimes, a subject discussed further in a later section.

Other lifetime distributions exist. However, the preceding cover the overwhelming majority of common parametric distributions used to model lifetimes. Further, while all the distributions described have mathematical and contextual justifications for their use in specific contexts, these are not always of utmost importance. Historically, the fact that a distribution provides an accurate fit for the data has been an overriding justification in many applications[45]. Furthermore, it should be stressed, regardless of which distribution is selected to model lifetimes, its adequacy should be checked. A topic which will be returned to in the following chapter.

Truncation and Censoring

Truncation and censoring are defining elements of time-to-event models. The functions that have been described so far have assumed an environment where the waiting time until an event is fully observed. In most real-world applications, however, it is impractical to wait until all systems in the population fail. Therefore, time-to-event models usually exist in incomplete information. Truncation and censoring provide mechanisms to formalize how that state of limited information affects the model. The following section describes how truncation and censorship affect data, provide several standardized schemes for censorship and how the phenomena can be expressed mathematically.

Truncation refers to when values outside a particular bounds are entirely omitted. Simply put, in a truncated data set, systems are only observed if they have survived up to a particular point in time. Truncation is less common in reliability contexts but does still occur. For example, if systems' lifetimes were recorded only if they had failed after a particular date. Then those systems that had failed prior to that particular date would not be recorded and thus truncated[31].

Censoring refers to when values are only known to occur within a particular range. In time-to-event analysis it is common to examine events that occur in the future. As a result, the exact timing of an event may not be known. Yet, its the direction from the observed value, is known. This observation still provides partial information about the timing of an event but not as much as a fully observed event.

The difference between censorship and truncation can be viewed as the difference between known-unknowns and unknown-unknowns. If truncation exists in a data set, observations are omitted from that data set and it is impossible to determine if those observations included events or not. With censoring, the source of the observations is retained, but there is uncertainty as to the exact timing of the observation. Both have detrimental effects on model fit, but the effect of truncation results in substantially greater bias.

There are different types of censoring depending on which segment of the data is missing. If the event in question is missing because it is in the future, right-censoring is said to present. If an event has already occurred but the object was not observed at the time, left-censoring is said to present. Interval censoring occurs when an event is known to have occurred within a specific range of times, but its exact timing is not known.

Censoring is also categorized depending on the mechanism leading to observations being censored. Different schemes exist, mainly from the literature on clinical trials, which define mechanisms where censorship occurs. This includes, if the event occurs after a particular end date (Type I) or if it occurs after a specific number of events have been observed (Type II). However, more common, is random or non-informative, censoring. As the name suggests, this is when censoring occurs randomly and is independent of failure times.

Censoring can be operationalized by augmenting the underlying distribution of the time until our well-defined event, T. First, a binary random variable d is defined. This variable takes on a value depending on what is observed at time t. If at time t the event has occurred, then d takes on the value of one. If at time t the event has not yet occurred and is therefore censored, d takes on the value of zero. With the introduction of d it is clear that T is not being directly observed. Rather, another distribution which contains

censoring times C is interfering. What is being observed is instead the minimum of both the failure times and the censoring times, which we denote as Y.

$$Y_i = \min(T_i, C_i) \qquad \text{and} \qquad d_i = \begin{cases} 1 \text{ if } T_i \leq C_i \\ 0 \text{ if } C_i < T_i \end{cases}$$

This provides a route to expressing what is actually being observed within a time-to-event data set, which is a combination of two distributions, one of real lifetimes, the other of censoring.

Hazard Modeling with Covariates

Hazards provide the means by which to assess the risk status of any particular system in the population over any given interval. This makes it an ideal metric through which to model the probability of failure. The constructions discussed so far have had an implicit assumption. Namely, that the lifetime distribution T arises from systems under identical conditions. Effectively, this implies that the population from which these lifetimes are derived is homogeneous. In practice, it is obvious that not all systems will be subject to the identical risks and conditions. Systems will have distinct manufacturers and locations, not to mention being exposed to an assortment of risks due to different operating conditions. However, to enable this diversification of lifetimes, a method to integrate covariates that formalizes this diversity is needed. The following section describes the Multiplicative Hazard Model which provides a means to express this diversity, and demonstrates several extensions to the model to allow for more complex covariates.

To incorporate covariates into the model of lifetimes, a function, $g(\cdot)$, that permits offsets for each individual system, i, is required[11][68].

$$h_i(t|\mathbf{x}_i) = g(t, \beta_1 x_1 + \cdots + \beta_m x_m)$$

This function should alter the hazard by a given vector of covariates, $\mathbf{x}_i = (x_1, \ldots, x_n)$, for each individual system, and a shared vector of coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$. Together, this is a linear model, $\boldsymbol{\beta}^T \mathbf{x}_i = \beta_1 x_1 + \cdots + \beta_m x_n$.

There are numerous routes to finding the function $g(\cdot)$. Yet, it is important to keep in mind the constraint on the original hazard. Any hazard must be positive, this means that the output of $g(\cdot)$ must also be positive. There is nothing to prevent the linear model from producing a negative output. However, a simple logarithmic transformation can prevent this outcome.

$$\begin{aligned} \ln(Y) &= \boldsymbol{\beta}^T \mathbf{x}_i \\ Y &= \exp(\boldsymbol{\beta}^T \mathbf{x}_i) \end{aligned}$$

Once the positive output hazard is ensured, there remains the question of the functional form of $g(\cdot)$. The most common solution is to treat the linear model as acting multiplicatively on a shared baseline hazard, $h_0(t)$.

$$h_i(t|\mathbf{x}_i) = h_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i)$$

The baseline hazard, $h_0(t)$ is the common hazard when all covariates are zero, as $\exp(0) = 1$. In this way, it is the considered the baseline as it occurs prior to considering any additional variables[39]. Note, the linear predictors do not contain an intercept term and rather begin with $beta_1 x_1$. This is because the baseline hazard already acts as an intercept for the model. The absence of the effect of the covariates would simply be the baseline hazard at that time.

This is the simplest form of the Multiplicative Hazard Model[5]. Whereas in a standard linear model, the estimated parameter value implies a linear change in every individual's fitted value, in this Multiplicative Hazard Model, a change in an estimated parameter implies a proportional shift in each individuals hazard at all time points[46].

There is nothing deterministic about the use of multiplication as the form of $g(\cdot)$. The linear model could have just as easily been added rather than multiplied. There exists an entire class of additive hazard models[43]. However, they are far less common in practice, mainly due to difficulties fitting these models[7]. This lack of application has led to a smaller quantity of available literature.

This multiplicative construction separates the baseline hazard, $h_0(t)$ from the linear model, $\boldsymbol{\beta}^T \mathbf{x}_i$. This allows for the baseline hazard and linear model to be estimated separately of each other. This greatly simplifies the fitting process. Furthermore, it allows for the comparison of the effect of any set of covariates through a proportion, known as a hazard ratio.

$$\frac{h(t|x_1)}{h(t|x_2)} = \frac{h_0(t)\exp(\boldsymbol{\beta}^T x_1)}{h_0(t)\exp(\boldsymbol{\beta}^T x_2)} = \exp(\boldsymbol{\beta}^T(x_1 - x_2))$$

In the above ratio, it is clear that time, t, along with the baseline hazard, $h_0(t)$, are factored out of the final construction. In this form, the hazard is not dependent on time, just on the effect of the covariates. This specific form of the Multiplicative Hazard Model is known as the Proportional Hazards Model.

The feature that allows for time to be factored out of the model is also a limitation. In separating the baseline hazard from the linear model, a hard assumption is made about the nature of the covariates. Specifically, that they must be time-invariant, or remain static through time. This does apply to many types of covariates that can specify the level of risk to which a system is exposed, like its location, manufacturer or whether it is part of a control group.

Fortunately, feature engineering methods can be used to construct variables that are time-invariant from ones that are time-dependent in raw data. This allows for any variables to be used within these models. Furthermore, it is possible to extend the Multiplicative Hazard model to explicitly include both time-dependent effects and time-varying covariate[15]. However, due to the complexity of the model fitting mechanisms of these constructions, especially with Frailties, these are beyond the scope of this text.

[5] Also known as the Cox Regression Model

Frailty

So far, all the models described have implicitly assumed that given a set of covariates, the resulting lifetimes are independent and drawn from the same underlying distribution with the same parameters. In practice, such an assumption, while useful, is rarely realistic. Thus, there is a need to extend the model further to allow for correlation between observations. This is done through the introduction of a Frailty into the model.

A Frailty[6] is an unmeasured random effect that is incorporated into a hazard function to account for heterogeneity in the population[32]. This unobserved random quantity impacts the hazard function multiplicatively.

$$Z \cdot h(t)$$

At its core, the Frailty provides a means by which to account for the fact that data often contains groupings or clusters of similar observations even if not explicitly modeled, leading to dependence among observations. This dependence implies some unknown correlation structure within and between these groups. The inclusion of a Frailty provides a method to model this dependence. It assumes conditional independence among lifetimes given the Frailty. This shared Frailty is the source of dependence within a group[77]. For example, inverters clustered within solar parks will exhibit similar lifetimes as a result of being exposed to similar conditions. Furthermore, inverters of a similar type or manufacturer are also likely to share attributes that lead to correlations in their lifetimes. This provides valuable information about the health of units within these groupings allowing for more accurate prediction of their lifetimes.

Yet, much of these similar conditions are unlikely to be fully captured by the covariates in the model. This omission occurs when relevant covariates cannot be observed or are too costly to observe. It is impossible to collect all the risk factors that contribute to failure. There is rarely awareness of all contributing factors, either because their relevance is not known or because there is a lack of means to measure them. This is especially the case for complex electronic systems whose components interact with one another as well as the environment. The cost of observation is of particular relevance to the photovoltaic domain, as the inclusion of relevant data is limited by financial constraints.

The form of the frailty determines how correlation structure is modeled. A Frailty is introduced into the Multiplicative Hazard Model as follows:

$$h_i(t|\mathbf{x}_i, Z_i) = Z_i \cdot h_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i)$$

Here, Z, is defined as a non-negative random variable, with some distribution, $\mathfrak{g}(\cdot)$, varying across the population[77] with Z_i being the realization of the Frailty for each individual system. Generally, the Frailty distribution is standardized, resulting in an average value equal to one, $\mathrm{E}[Z] = 1$, and a variance parameter that is estimated from the data, $\mathrm{Var}(Z) = \theta$.

It is important to distinguish between the frailty itself and the random effect, b_i^t:

$$h_i(t|\mathbf{x}_i, Z_i) = h_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_i + b_i^t z_i)$$

[6] Also known as a mixed/random effect, longevity factor, susceptibility or liability. When introduced into the model, it makes a mixed hazard model, or a hierarchical hazard model

Where $Z_i = \exp(b_i^t z_i)$ and clearly when $b_i = 0$ then $Z_i = 1$. This illustrates that the multiplicative effect the Frailty intends to model can alternatively be viewed as an additional covariate in the linear model.

As the average value of the Frailty is set to one, divergences from the average characterize the effect on the hazard. Individual systems with Frailties greater than one will have a higher than average hazard, implying they are more frail. Meanwhile, individual systems with Frailties less than one will have lower than average hazard implying they are less frail. Those that are more frail will die earlier than those that are less frail.

Estimating a regression model without the introduction of random effects can bias the resulting predictions if individual observations are not truly independent. In time-to-event data, this issue is compounded by temporal effects. Individual systems that are more robust will almost certainly live longer and produce more data. The population hazard declines as a result of high-risk individuals failing, but the individual hazard may continue to increase. This systematic observation of the most robust individuals will skew the computation of the average hazard. Thus, an estimate of the individual hazard rate without taking into account unobserved Frailty will underestimate the hazard function to an increasingly greater extent as time goes by[1].

The estimation of the variance parameter, θ, describes the degree of diversity in baseline hazard of a population. A large θ implies a high degree of heterogeneity as lifetimes deviate heavily from the average value of one. Conversely, a small θ implies the homogeneity of lifetimes as each one will be heavily concentrated near the average value.

In the above construction, the Frailty is introduced in such a way that the entire population of systems is covered by a single distribution. In practice, it is more useful to estimate a Frailty for each distinct group within the population, as follows:

$$h_{ij}(t|\mathbf{x}_i, Z_j) = Z_j \cdot h_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_{ij})$$

Here, a Frailty of a collection is introduced and each distinct group j is composed of a total of k distinct groups. A random variable, represents the collection, $Z_j = (z_1, \ldots, z_k)$. A collection is defined as a set of groups that all share a common characteristics. To be properly defined, a collection should be mutually exclusive and exhaustive given the data. For example, there can only be a finite number of parks within which inverters reside. Each park makes up a distinct group, and all of the parks make up the collection. With this structure, it is possible to estimate the Frailty, and thus the level of heterogeneity for each group. The interpretation of the Frailty and its variance are identical to what has been described previously, except localized to each distinct group. This can be extended further to allow for as many different collections as is required, simply by introducing additional Frailties into the model multiplicatively. It should be noted, the multiplicative introduction of further Frailties implies independence between lifetimes of different collections. In fact, the lifetimes of individual systems are conditionally independent given the vector of Frailties.

Any distribution which is positive and possesses a mean can be set to one, can be used as the Frailty distribution, $\mathfrak{g}(\cdot)$. The most commonly used distribution is the Gamma. It is utilized due to its flexibility in modeling positive outcomes as well as the ease with which it allows for the expression of all required formulas[77]. The density of the Gamma is given as follows:

$$f(t) = \frac{\lambda^k t^{k-1} \exp(-\lambda t)}{\Gamma(k)} \qquad (k > 0, \lambda > 0)$$

As stated earlier, the goal here is to fix the average Frailty to one and estimate the variance. This can be done the following way:

$$\mathrm{E}[Z] = \frac{k}{\lambda} = 1 \qquad (k = \lambda)$$

$$\mathrm{V}[Z] = \frac{k}{\lambda^2} = \frac{1}{\sigma^2} \qquad (k = \lambda)$$

The restriction of $k = \lambda$ is made to ensure that the average Frailty is equal to one. Through this restriction we note that the variance is now, by definition, $\sigma^2 = \frac{1}{\lambda}$. This leads to the conditional density of the Gamma distributed Frailty:

$$f(z) = \frac{1}{\Gamma(\frac{1}{\sigma^2})} \left(\frac{1}{\sigma^2}\right)^{\frac{1}{\sigma^2}} t^{\frac{1}{\sigma^2}-1} \exp(-\frac{t}{\sigma^2})$$

This density can be transformed into a unconditional survival function and consequently the into the unconditional density and hazard using the following formulas:

$$f(t) = \frac{h_0(t)}{(1 + \sigma^2 H_0(t))^{\frac{1}{\sigma^2}+1}}$$

$$S(t) = (1 + \sigma^2 H_0(t))^{-\frac{1}{\sigma^2}}$$

$$h(t) = \frac{h_0(t)}{1 + \sigma^2 H_0(t)}$$

The corresponding hazard and cumulative hazard can be replaced with the appropriate lifetime distribution. For example, let $h(t) = \frac{\alpha}{\gamma}\left(\frac{t}{\gamma}\right)^{\alpha-1}$ be Weibull distributed baseline hazard with a Frailty following a Gamma distribution $Z \sim \text{Gamma}(\frac{1}{\sigma^2}, \frac{1}{\sigma^2})$. Thus, the unconditional survival and hazard functions are given by the following expressions:

$$S(t) = \left(1 + \sigma^2 \left(\frac{t}{\gamma}\right)^{\alpha}\right)^{-\frac{1}{\sigma^2}}$$

$$h(t) = \frac{\frac{\alpha}{\gamma}\left(\frac{t}{\gamma}\right)^{\alpha-1}}{1 + \sigma^2 \left(\frac{t}{\gamma}\right)^{\alpha}}$$

Where σ^2 represents the variance of the Gamma distributed Frailty.

There are several important issues that should be kept in mind when introducing Frailties to time-to-event models. A Frailty is assumed to be constant over time. If the random variable Z is introduced as a Frailty into the Multiplicative Hazard Model, then

the estimation of Z is fixed for all time points. However, this does not mean that Frailties experience no temporal effects. This can be seen when the conditional expectation and variance of the Frailty Z are derived:

$$\mathrm{E}[Z|\mathbf{x}, T > t] = \frac{1}{1 + \sigma^2 H_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x})}$$

$$\mathrm{V}[Z|\mathbf{x}, T > t] = \frac{\sigma^2}{(1 + \sigma^2 H_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}))^2}$$

Two things of note can be understood from the above. First, individuals dying at time t will have a higher mean Frailty compared to survivors. This is because the cumulative hazard in the denominator increases as time goes on. Second, the variance of Frailty declines over time. This implies that early failures reduce the heterogeneity of of the population over time. However, the ratio between the average Frailty and its variance remain constant over time implying that the population does not become more homogeneous relative to the average in time[77].

The introduction of a Frailty, at any level, will invalidate the Proportional Hazards property[39]. The property, explained in the previous section, allows for the a comparison between any set of covariates through a simple proportion. However, with the introduction of the Frailty, any sets of covariates now require the computation of a temporal effect, such that:

$$\frac{h_1(t)}{h_2(t)} = \frac{1 - \sigma^2 H_0(t)}{1 - \sigma^2 H_0(t) \exp(\boldsymbol{\beta})} \exp(\boldsymbol{\beta})$$

The ratio of specific population hazards is generally not time-invariant. This can be seen from the above equation, the hazard ratio is a decreasing function, unless there is no difference between groups or the Frailty is not relevant. The reason for this are, again, the alteration of the hazard rate in the population as high-risk individuals fail earlier leading to a lower average hazard rate over time.

There are significant advantages to using a parametric baseline hazard when introducing Frailties into a Multiplicative Hazards Model. The most important consequence of such a combination is that it enables the explicit description of the evolution of a Frailty over time. Without a parametric form it is difficult to describe how

In a traditional context, the values of Z_i are not estimated for each individual system within the population. The reasoning for this is relatively simple. An estimate of a Z_i for each individual system, i, would require N parameters. This would result in more parameters in the model than there are observations, leading to an over-saturated model. Furthermore, the estimation of individual Frailties are usually less important than group Frailties and variances. These provide useful estimates of the degree to which these groupings affect survival times, as well as identify patterns among those groups. That said, this traditional restriction is loosened in the context of Bayesian estimation where an estimation for Z_i can be sampled.

Model Likelihoods

The likelihood is a function of the parameters of a statistical model given data. Its formulation is the means by which the appropriate mathematical abstraction is selected after data has been observed. In this section, likelihoods for the models developed so far are formulated. First in the simplest cases, then with the model additions.

Generally, a parametric model is fit by finding the maximum values for a set of parameters, θ, of its probability density function, $f(\cdot)$, given a vector of data, $T = (t_1, \ldots, t_n)$. This process is encapsulated in Maximum Likelihood Estimation (MLE):

$$\arg\max_{\theta} \mathcal{L}(\theta|T) = \arg\max_{\theta} f(T|\theta)$$
$$\mathcal{L}(\theta|T) = \Pr(T|\theta) = \prod_{i=1}^{n} f_\theta(t_i)$$

In the time-to-event context, there is an added complication. As noted earlier, this type of data often features censoring and truncation. When non-informative right-censoring is present, it is required to extend the likelihood. The data, T, is now input as a pair, (y_i, d_i). The Y random variable is defined as the minimum of the actual and censored lifetime. It represents the observed lifetimes found in the data. A binary random variable, d, is introduced to make explicit when censoring occurs. The likelihood is altered as follows:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(y_i)^{(d_i)} S(y_i)^{1-d_i}$$

The probability density function, $f(y_i)$, of the parametric distribution is still present in the above equation, much like in the standard likelihood. However, its contribution is now controlled by the d random variable. When the variable d equals one, it implies a failure event and the likelihood is evaluated in the general way. However, when d is zero, censoring occurs and the likelihood evaluates instead the survival function, $S(y_i)$, which implies the object has survived up to that particular point in time. The combination provides information about both observed events and about yet to be observed events.

Given the identities found at the beginning of this chapter, it is possible to transform the joint likelihood to make use of hazards. Such that the likelihood becomes:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} h(y_i)^{d_i} \exp(-H(y_i))$$

This generally provides easier functions to work with when constructing the likelihood. It should be noted that hazard estimation is restricted here to only include parametric forms. It is incredibly common to not make parametric assumptions about the baseline hazard. There is a large body of research into non-parametric baseline hazard estimation, including the Kaplan-Meier and Nelson-Aalen estimators. If contrasting the effect of covariates is the predominant purpose of the model, semi-parametric or non-parametric methods are often preferred. However, the structure of the baseline hazard has substantive implications for the ability to understand the model and predict. This is especially the

case when Frailties are involved as parametric hazards aide in understanding the evolution of group hazards over time. Furthermore, non-parametric methods lack the ability to forecast future hazard rates beyond the last failure time, as any value beyond the last time is simply undefined. That said, the parametric form is not without its price, namely it is an assumption about the evolution of lifetimes which must be verified to be used.

The likelihood for the censored model can be extended to include covariates. First, recall the Multiplicative Hazards Model:

$$h_i(y|\mathbf{x}_i) = h_0(y)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)$$

In this case the data is input as a triple, $(y_i, d_i, \mathbf{x}_i)$ as the covariates are also introduced. The likelihood for this function is as follows:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n}\left(h_0(y_i)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)\right)^{d_i}\exp(-H_0(y_i)\exp(\boldsymbol{\beta}^T\mathbf{x}_i))$$

The introduction of a univariate Frailty into the Multiplicative Hazard Model is straight-forward. As stated earlier, the Frailty model can be generally stated as:

$$h_i(y|\mathbf{x}_i, Z) = Z_i \cdot h_0(y)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)$$

Thus, the likelihood with univariate Frailty and non-informative right censoring simply expands to:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n}\left(Z_i \cdot h_0(y_i)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)\right)^{d_i}\exp(-Z_i \cdot H_0(y_i)\exp(\boldsymbol{\beta}^T\mathbf{x}_i))$$

As can be seen from the above, a product is now required to handle the additional, k, Frailty terms. In each extension the effect is simply the multiplication of an additional term, or set of terms, on the baseline hazard.

If a Gamma Frailty is used, and the parameters, Z are integrated out, the likelihood of the univariate Frailty becomes:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n}\left(\frac{h_0(y)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)}{1+\sigma^2 H_0(y)\exp(\boldsymbol{\beta}^T\mathbf{x}_i)}\right)^{d_i}(1+\sigma^2 H_0(y)\exp(\boldsymbol{\beta}^T\mathbf{x}_i))^{-\frac{1}{\sigma^2}}$$

The extension to shared Frailty adds an additional complication. Specifically, for each cluster a separate frailty distribution must be fit. As right censoring is at play, this vastly complicates the model likelihood. For, n, clusters, j, of size, n_i,, such that, $i \ldots, n$, the shared frailty becomes[77]:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n}\frac{\Gamma(\frac{1}{\sigma^2}+\delta_i)\prod_{j=1}^{n_i}(h_0(t)\exp(\boldsymbol{\beta}^T\mathbf{x}_{ij}))^{d_{ij}}}{(\frac{1}{\sigma^2}+\sum_{j=1}^{n_i}H_0(t)\exp(\boldsymbol{\beta}^T\mathbf{x}_{ij}))^{\frac{1}{\sigma^2}+\delta_i}\sigma^{\frac{2}{\sigma^2}}\Gamma(\frac{1}{\sigma^2})}$$

Where $\delta_i = \sum_{j=1}^{n_i} d_{ij}$ or the count of events within each cluster and the Gamma function is $\Gamma(t) = \int_0^\infty x^{t-1}\exp(-x)dx$. Taking the log of the above, we get the log likelihood[18]:

$$\ell(\theta) = \sum_{i=1}^{n} \left[\delta_i \ln(\sigma^2) + \ln \Gamma \left(\frac{1}{\sigma^2} + \delta_i \right) - \ln \Gamma \left(\frac{1}{\sigma^2} \right) - \left(\frac{1}{\sigma^2} + \delta_i \right) \ln \left(1 + \sigma^2 \sum_{j=1}^{n_i} H_0(t) \exp(\boldsymbol{\beta}^T \mathbf{x}_{ij}) \right) + \sum_{j=1}^{n_i} d_{ij} \left(\boldsymbol{\beta}^T \mathbf{x}_{ij} + \ln h_0(t) \right) \right]$$

As we can see this likelihood is incredibly complex. Yet, the Gamma Frailty is one of the few distributions that is capable of being fit in this manner, as it has a closed form solution for the log likelihood. Yet, there is another way this can be done without explicitly deriving the log likelihood. Specifically, by introducing the Frailty, Z_j as a parameter on the original likelihood of the Multiplicative Hazard Model and allowing sampling to directly estimate it. Thus, the shared Frailty retrns to:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \prod_{j=1}^{n_i} \left(Z_j \cdot h_0(y_{ij}) \exp(\boldsymbol{\beta}^T \mathbf{x}_{ij}) \right)^{d_{ij}} \exp(-Z_j \cdot H_0(y_{ij}) \exp(\boldsymbol{\beta}^T \mathbf{x}_{ij}))$$

The added benefit of directly sampling from such a likelihood construction is that the parametric form is no longer restricted to the Gamma distribution. Any positive distribution, such as the log-normal or inverse Gaussian, can be used to model the Frailty parameter. Furthermore, the restrictions imposed by the use of the Gamma as the Frailty distribution, such as a uniform correlation structure can be discarded. Of course nothing is free, and the price paid for such an extension is that the model will take substantially more computer time to fit correctly. However, this provides a powerful means to further extend the model.

There are many technical details which are required to properly fit a Frailty model. Most of these are specialized and are far beyond the scope of this text. The reader is directed to the "Extended Reading" section at the end of this chapter for more details on this front.

Model Estimation

Model estimation can be done in a number of ways. However, as model complexity increases with the addition of covariates, random effects or other features, increasing difficulties in using traditional methods of model fitting are encountered. Many of these issues arise from a lack of closed form solutions for the required mathematical constructions needed to fit these models. Bayesian methods provide a means to circumvent some of this difficulty through the use of sampling rather than explicit formulation to derive the model output. In this section we examine the Bayesian paradigm and provide some intuition as to the model fitting technique used in the remainder of the text, Hamiltonian Monte Carlo (HMC).

In the preceding section, likelihoods for the various models were presented. It was stated that these parametric models could be fit by finding the maximum of the likelihood

function. While in simple cases this is practical, as model complexity increases this becomes considerably more difficult. The introduction of random effects, like shared Frailties, pose an especially large challenge as closed form solutions for the models either become too complicated or cease to exist. In the case of Gamma Frailties, these closed form solutions still exist but are remarkably complex. The final equation from the previous section is an affirmation of that fact. However, if the model were to be extended further to introduce a different parametric distribution for the shared Frailty, the likelihood would lack an explicit formulation. For example, log-normal Frailties no longer have a closed form solution for a likelihood that factors out the Frailty. In these cases, Expectation-Maximization, Adaptive Gaussian quadrature or Markov Chain Monte Carlo (MCMC) are some of the few available numerical optimization methods that can be employed[77]. While the exact process for fitting such model extensions is beyond the scope of this text, it is important to provide the route by which the model could be extended further. For that, a flexible fitting mechanism is required. In this case, Hamiltonian Monte Carlo, which is a special case of MCMC is used.

In the Bayesian paradigm, the posterior distribution encompasses the credibility of the parameter values, of a particular model. Unlike in traditional statistics, the vector of unknown parameter values, θ, is assumed to be random variable with its own distribution as well as a prior distribution, $\Pr(\theta)$[33]. Inferences concerning the parameter values are based on the posterior distribution, given by:

$$\Pr(\theta|D) = \frac{\mathcal{L}(\theta|D)\Pr(\theta)}{\int_\theta \mathcal{L}(\theta|D)\Pr(\theta)d\theta}$$

Where D is the observed data. It can be seen from the above equation, that the posterior probability is proportional to the likelihood multiplied by the prior.

$$\Pr(\theta|D) \propto \mathcal{L}(\theta|D)\Pr(\theta)$$

This is because the denominator is a normalizing constant:

$$\int_\theta \mathcal{L}(\theta|D)\Pr(\theta)d\theta$$

This normalizing constant is interesting because in Bayesian models, it often lacks a closed form solution and has to be estimated through an alternative process. This produces a similar situation to the one found in the time-to-event models. When the closed form solution is not available, similar methods as those used for fitting Bayesian models become useful. However, even when closed form solutions are available, the Bayesian method of fitting may prevent the need to manually integrate a complex likelihood. Additionally, features like shared Frailties can be re-contextualized as Bayesian hierarchical models. This forestalls the requirement for asymptotic arguments required to compute model extensions. In contrast, in the Bayesian context, these features are simply by-products of sampling from the posterior and their calculation becomes substantially simpler[33].

The most common approach for describing a posterior distribution make use of MCMC methods. These methods generate representative samples from the posterior distribution. The earliest method for generating representative samples, also known as the Metropolis-Hastings algorithm[49], proceeds as follows: First, an arbitrary starting point along the distribution is selected and its value is recorded. This value is the posterior probability at

that point. Next, a candidate point is randomly selected around the starting point. If the value of the candidate point is greater than the starting point, then the algorithm moves to that new point. If the value of the candidate point is less than the starting point, then the the move is made probabilistically. The probability of the move is determined by the ratio of values between the starting point and the candidate point. Such that:

$$\Pr_{\text{accept}} = \min\left(\frac{\Pr(\theta_{\text{proposed}}|D)}{\Pr(\theta_{\text{current}}|D)}, 1\right)$$

Once the algorithm decides whether or not to move, the process begins again, with that position being the new starting point. This process is repeated a very large number of times, which creates a representative sample of the posterior distribution.

There have been numerous attempts to improve the efficiency of this simple algorithm. The largest inefficiencies arise in two areas. The tails of the distribution require a large amount of iterations to accurately map. This is because each step toward those tails has a very small probability of being accepted. For much the same reason, distributions with several local optima require a greater amount of iterations for the algorithm to escape these optima. Many of the attempts to improve the efficiency of this algorithm have tried to address these deficiencies. The most recent is Hamiltonian Monte Carlo (HMC), which was codified into the Stan statistical software package[8].

The central difference between the Metropolis-Hastings algorithm and HMC, is the mechanism for determining the proposal distribution[40].

After the algorithm selects a starting point, it has to randomly select a candidate point. These candidates are usually drawn from a symmetric distribution like a multivariate Gaussian. This fixed shape leads to candidates being selected regardless of where in the distribution the starting point is. As a result, candidates can just as easily travel away from the posterior mode as towards it. HMC uses a proposal distribution that differs based upon the current position. It alters the proposal by calculating the gradient, or the direction of change, of the posterior distribution. It then distorts the proposal distribution to match that gradient.

The process of selecting from candidates is also altered in HMC. Rather than simply selecting a point from the proposal distribution and evaluating it, HMC uses a 'momentum', around the candidate point. The area around the selected point is then allowed to be sampled given this momentum. The final point is selected based on the value that the point 'rolls' into[40]. The probability of acceptance takes into account this momentum when finally deciding on a move.

$$\Pr_{\text{accept}} = \min\left(\frac{\Pr(\theta_{\text{proposed}}|D)\Pr(\phi_{\text{proposed}}|D)}{\Pr(\theta_{\text{current}}|D)\Pr(\phi_{\text{current}}|D)}, 1\right)$$

As before, this process is repeated a multitude of times until the posterior probability distribution is approximated. From this approximation, it is possible to extract the features necessary to enable model fitting. These alterations make HMC more computationally costly, as the gradient is required to be calculated. However, it is also considerably more efficient as fewer samples are required to build a representative posterior distribution.

Numerous additional technical details are required to properly understand HMC which are not covered in this text, such as No U-Turn sampling, step size and duration tuning. Fortunately, software will optimize these particularities for a user. If the reader

is interested in these topics, the 'Extended Reading' section provides some references of interest.

Extended Reading

For additional treatments of the topic, several notable sources can be accessed. For a comprehensive mathematical treatment of classical survival analysis, the reader is directed to Klein and Moeschberger[38]. For a mathematical treatment of survival analysis based on counting processes, the reader is directed to Andersen[2] and Aalen, Borgan and Gjessing[1]. If the reader is interested in a pedagogical viewpoint on survival analysis or model computation, the reader should explore Kleinbaum and Klein[39] and Tableman and Kim[68]. For a treatment of Bayesian time-to-event analysis including extensions to this model, the reader is directed to Ibrahim, Chen and Sinha[33]. For a detailed description of Frailty Modeling, including detailed mathematical derivations of incrementally complex models, Wienke[77], as well as Duchateau and Janssen[18] should be explored. For more details on Hamiltonian Monte Carlo, the reader is directed to Kruschke[40], Gelman et al[23]. For a technical review of Bayesian estimation, Marin and Roberts[44] are suggested.

Analysis and Application

This chapter outlines the heuristic for building a predictive maintenance process. It begins by restating the goal of predictive maintenance in precise terms. It then explores data management and feature engineering. The quality of a predictive maintenance process in the photovoltaic domain largely depends on the ability to extract meaningful covariates from the small amount of relevant information. How data should be combined and what methods can be used to improve the quality of covariates is discussed. Then, the data simulation process is returned to, this time with its mathematical underpinnings fully exposed. A comprehensive structure of the simulated data set is provided. Then, a brief introduction to Stan, the probabilistic programming language, is given. This includes a detailed set up of the time-to-event models described in the previous chapter. Finally, the model fit, evaluation and performance are presented.

The Goal

The goal of a predictive maintenance process in the photovoltaic domain is to create a system that allows for the correct allocation of maintenance resources regarding inverters with the greatest potential for failure. Formally, this can be seen as creating an sequence of inverters ranked by a score function, V, at a given time. Such that:

$$V_{(1)}(t) > V_{(2)}(t) > \dots V_{(n)}(t)$$

Where the time point, t, is fixed at the current moment for prediction. It should be noted, the absolute value of the score is of secondary importance in this process. If maintenance resources are allocated as part of a continuous service it does not matter if the inverter with the highest score is likely to fail within the next day, week or month. It is still the most likely to fail and should still be the inverter to which the maintenance process is applied to first. Therefore, an exact determination of risk is secondary to the action it prompts. This equates time-to-event analysis with rank-order prediction, where the order of failures, and thus the order of maintenance activities are the central goal.

Feature Engineering

Feature engineering is sometimes considered the dark art of the statistical modeling process. Its informal nature makes it difficult to find a general consensus as to what the concept actually means within academic literature[81]. Loosely speaking, feature engineering is the art of creating covariates that conceptually embody aspects of a phenomena or object of observation. In this case, it refers to the task of finding and encoding historical information that can describe a photovoltaic inverter's health at any given moment.

While a multitude of effort is expended generating novel models which explain and predict better, considerably less attention is focused on what inputs are useful in these models. No domain ever comes with all the ideal features built into existing data sets. Thus, a degree of skill is required to extract and encode what is important. In a practical setting, like the development of a predictive maintenance process, the proper application of feature engineering is the difference between an effective or ineffective method[80]. Part of this limited attention stems from the fact that feature engineering is often domain-specific and cannot be adequately generalized. That said, there are themes that can be used to guide the process. The two primary sources of features are subject matter expertise and experimentation. Subject matter expertise can be further subdivided into model-specific and domain-specific insights.

An understanding and awareness of the details and limitations of the model is paramount to being able to create covariates that act as effective inputs. This requires subject matter expertise, which arises from knowledge of the functional attributes of the applied model, such as that presented in the previous chapter. Such knowledge can avoid the negative consequences of using data that may be suboptimal in combination with certain models. For example, when introducing maintenance logs as a time-dependent covariate, it is not sufficient to simply provide a binary variable that records whether an inspection occurred on a specific date. The functional form of the model implies such a formulation is unlikely to serve as a sufficiently strong signal for the effect of regular inspection on lifetimes, as it lacks variability. However, the number of days since the last inspection is more likely to have a substantive effect. The functional structure of the model reveals these conditions and should be kept in mind when constructing covariates.

Domain-specific subject matter expertise is an inescapable necessity for effective feature engineering. While there continues to be a drive toward greater automation in determining the relevance of covariates, human decision continues to remain paramount. In this area, domain-specific subject matter refers to an understanding of the contributing events that lead to failure in photovoltaic inverters. This includes knowledge about the various failure modes of an inverter and their origins. This includes such sources as high voltage, extreme temperatures, water condensation, the lack of robust software by certain manufacturers and others factors. It also includes knowledge about the hardware itself and which components are under stress and contribute to failure, such as capacitors, semiconductor switches and housing materials[21]. Domain-specific knowledge provides the necessary scope with which to limit the search for features. It is often best generated from continuous interactions with front-line workers on the system in question.

Experimentation is the last source of feature engineering and is meant to complement rather than substitute subject matter expertise. As its name suggests, this is where features are created and then tested to see whether or not they improve model performance.

This includes experimenting with feature encodings, such as making use of power transformations as well as variable discretization, standardization and normalization. All of these changes can potentially improve model performance. It may also include generating new features based on an assumption of effect. For example, it may be wise to include a feature that encodes days with both low temperature and low humidity which can cause printed circuit boards to crack. Then this covariate should be tested to see if it improves predictive performance.

As feature engineering is more an art-form than a science, there is no single correct method for performing the task. Despite the lack of a clear heuristic, there are some guidelines for introducing features in time-to-event models. An important condition to keep in mind is that time-to-event models have difficulty in dealing with noisy variables across small intervals. The purpose of these models is to determine the contribution of covariates to the rate of failure. However, the introduction of stochastic variables, such as daily temperature, may not have that desired effect. This is because the effect of temperature on an inverter is only relevant in certain extreme cases over time and not in any one case. The effect of a single day of extreme temperature is unlikely to result in a failure on that day. In effect, this violates the association that the model demands between covariate value and inverter status. This can be partially offset by the use of rolling aggregates, that provide an average temperature over a specific intervals. However, by definition, these extreme events are rare, and will likely be smoothed out in these rolling aggregates. Furthermore, this does not account for compounding effects. A better solution is the use of monotonically increasing variables like a count of extreme temperatures across a specific interval. Of course, such a feature requires a definition for 'extreme', one which should be subject to experimentation on a validation set. Yet, it is important to realize that time-dependent variables require trajectories to be utilized effectively for predictive purposes.

Data Management and Simulation

Raw data can rarely be input into a model as-is. The first chapter broadly examined the standard data sources for a predictive maintenance process as well as provided a general outline of the simulated data. This section returns to these topics and examines the structure of the input data. It then turns its attention to the process of data simulation, introducing the quantile function and the inverse cumulative hazard. Finally, it presents the data used in the subsequent analysis.

Data required for a predictive maintenance process often comes from several different database tables meant to record different aspects of a collection of systems. This includes databases that document system attributes, maintenance and failure history and condition monitoring and usage patterns. Conversely, time-to-event models, require a form of attribute-value data. This data format can be roughly thought of as tabular data. Each row is a system, and each column encodes a covariate, with one column containing the failure status of the system. The input data then provides a table of the last observed value of each system and their respective covariates. These values can then be transformed into a table which can be fed into Stan.

Figure 2 gives an illustration of how data should be structured on input. The suffix, 'l30d' takes the count in the last thirty days of operation of a particular value, such as

Covariate/Feature	Description
time	observed time
censor	censoring indicator
id	unique identifier for inverter
park	unique identifier for park
type	inverter type: central string or micro
kWh_l30d	average kwh in last 30 days standardized
weather_l6m	count of extreme weather events in last 6 months
err_code01_l30d	count of minor errors in last 30 days
err_code02_l30d	count of major errors in last 30 days
repair_days	days since last repair

Figure 2: Covariates/Features of Simulated Data

'kWh' which encodes outlying energy output or 'weather' which encodes extreme weather events. The suffix, 'days', counts the number of days since an event, in this case the number of days since any repair was performed on the inverter.

Recall that the input for the models was structured as follows:

$$(Y_i j, d_{ij}, Z_i, \mathbf{x}_{ij})$$

In this case, the observation time is Y, the censor indicator is d, the park is the Frailty, Z, and the remaining variables make up, $\mathbf{x}$. It should be noted that this data is an example of what should arise after the feature engineering process is complete. That said, this is far simpler data than what would result from a real-world setting and is meant for illustration purposes. The goal is to produce a set of covariates that can adequately represent the failure mode of a device in order to enhance the understanding of the model not to necessarily replicate all the nuance of the real-world.

Data is simulated using inverse transform sampling. This method generates random values from any distribution using only standard uniform random values as inputs. Generally speaking, this enables the generation of values from any continuous distribution.

Briefly, let F be a continuous cdf. This guarantees that F^{-1} exists as a function from $(0, 1]$ to $\mathbb{R}$. If U is defined as a uniform random variable on the unit interval, $(0, 1]$ and $T = F^{-1}(U)$, then T is a random variable with the cdf, $F(\cdot)$. As the function of a random variable is also a random variable itself, the inverse of the cdf, $F^{-1}(U)$, known as the quantile function, is also a random variable[6]. This implies that, $F^{-1}(U) = T$ if and only if $U = F(T)$. Thus, all that is required to generate random values from any given distribution is the quantile function and uniform random variables.

In this case, there is a desire to generate survival lifetimes. In the previous chapter, the cdf of a general lifetime distribution was given as follows:

$$F(t) = 1 - S(t) = 1 - \exp(-H(t))$$

Due to symmetry, a uniform random variable is defined on the unit interval for the survival function as well as the cdf. If $U \sim \text{Unif}[0, 1]$ then $(U - 1) \sim \text{Unif}[0, 1]$ as well. Thus:

$$U = \exp(-H(T)) \sim \text{Unif}[0, 1]$$

As long as all hazards are defined to be strictly positive, $h(t) > 0$, the cumulative hazard, $H(t)$ is invertible and the survival time can be expressed as:

$$T = H^{-1}(-\ln(U))$$

Fortunately, for all of the models presented thus far an inverse of the cumulative hazard exists. From this starting point it is possible to generate models with considerably more complex features.

For the Multiplicative Hazards Model with covariates, survival times are generated the following manner[5]:

$$T = H_0^{-1}(-\ln(U)\exp(-\boldsymbol{\beta}^T\mathbf{x}))$$

Where H_0^{-1} is the inverse of the baseline cumulative hazard function which is multiplied by the exponentiated effect of the covariates.

This simulation can be extended further to include the a Frailty term, Z_i, it then becomes[61]:

$$T = H_0^{-1}\left[\frac{-\ln(U)}{Z_i\exp(\boldsymbol{\beta}^T\mathbf{x}_{ij})}\right]$$

Of course, the precise model depends on the parameterization of the baseline cumulative hazard. As stated in the previous chapter, numerous different distributions can be used for modeling lifetimes and any of them can be used for this baseline cumulative hazard. In this case, a Weibull baseline hazard is employed. The resulting model with shared Frailties can therefore be simulated as follows:

$$T = \left[\frac{-\ln(U)}{Z_i\gamma^{-\alpha}\exp(\boldsymbol{\beta}^T\mathbf{x}_{ij})}\right]^{1/\alpha}$$

Where u is the realization of a uniformly distributed random variable, $U \sim \text{Unif}(0, 1]$ and α and σ are the shape and scale parameters of Weibull distribution, respectively.

Evaluating Model Performance

Time-to-event models focusing on predictive outcomes create a unique problem for evaluating performance. On one hand, the output of the model creates hazards, which are continuous and have a support on all positive real numbers $[0, \infty)$. In parametric models, these hazards are monotonically related to expected residual lifetimes. So an estimate of the difference between the hazard and observed remaining lifetime could be used to assess model performance. On the other hand, a determination of the exact failure time provides more information than is required for the task, not to mention censorship makes the real lifetime difficult to observe in many cases. Knowledge of the order in which failure occur is generally sufficient to schedule maintenance activities. This is especially true given the limited amount of information available in photovoltaic systems.

This section provides the basis for evaluating performance of time-to-event models for predictive outcomes. It begins by discussing the goal of model performance and addresses the difficulties that arise from applying traditional model evaluation techniques in the

time-to-event context. It then introduces the Concordance-Index (C-Index) as well as the Widely Applicable Information Criterion (WAIC) which are used for model discrimination and calibration. Finally, a general heuristic is provided for evaluating performance in this class of models.

Ideally, a well-performing time-to-event model should behave in a manner that generalizes the relationship between the covariates and the lifetime. After all, the purpose of any model is to discover the patterns that reveal the phenomena generating a class of data rather than any specific data set. Thus, a well-tuned model of lifetimes should apply to observations of new photovoltaic inverters just as well as it does to those already in operation. To assess this generality, it is important to ensure that the model performs well out-of-sample. Failure to do so can result in underfitting or overfitting, both of which are detrimental aspects of model performance.

Underfitting refers to when a model is unable to capture the complexity of the underlying phenomena. This is caused by the sin of omission, such as when important structural features or relevant covariates are not included in the model. As a result, the model is inflexible and will systematically bias results both in and out-of-sample. This is because underfitting results in an insensitivity to the structure of any sample. Overfitting refers to when random variation is treated as structure by a model. In an extreme case, overfitting can be thought of as data memorization. Thus the model memorizes the sample used to fit it. This makes it incredibly accurate in-sample. However, since new data is unlikely to have the exact same configuration as any single sample, it would result in poor out-of-sample, or generalized performance. Broadly speaking, the difference between underfitting and overfitting rests in the complexity of the model as well as the sensitivity of that model to the exact composition of the sample used to fit it[48].

As both underfitting and overfitting affect out-of-sample performance, it is important to utilize a metric that takes into account the generality of the model. One of the most common methods to do so in a predictive context is k-fold cross-validation. In the simplest version of the technique, the data is randomly split into k equal sized samples. Then, the model is fit on $k-1$ samples, with the remaining one being used to evaluate out-of-sample performance. This evaluation usually takes the form of mean-squared error (MSE) or mean-absolute error (MAE), both of which measure the distance of predicted values to the observed values in the k^{th} sample. This last sample is not used to fit the model thus acts as a replacement for out-of-sample data. An average of the MSE or MAE can then be taken to provide the general performance of any model.

Unfortunately, with time-to-event data, k-fold cross-validation becomes difficult to apply. The process has an implicit assumption that each observation is independent and therefore can be randomly split into k samples. In a time-to-event context, the sequence of lifetimes is of importance. As such, this method would result in historical information about a lifetime being randomly excluded from model fitting. The result would be a systematically underfitting of the baseline hazard. It would also increase deviations from the real hazard over time, as changes in initial conditions would result in greater deviations from the true hazard as time progresses.

Censoring also creates problems for this technique. It results in data that is severely unbalanced. Very few industrial settings feature an abundance of failures. Decades of reliability engineering has ensured that most industrial systems are not prone to widespread failure. Photovoltaic inverters are not excluded from this generalization[57]. As a result, the majority of observations will be non-events. While all observations may eventually

become events, as all systems eventually break down, it is highly unlikely in a production setting that the data will be balanced. Datasets where the majority of observations are non-events are the rule in time-to-event contexts, not the exception. As a result, the balanced comparison of predicted to observed lifetimes is difficult as that data makes up such a minor proportion.

In the time-to-event context, the Concordance Index (C-Index) can serve as a reasonable measure of the predictive performance of a model. It is commonly used in prognostic studies in the medical domain[70]. The C-Index is the conditional concordance probability measure of a lifetime and a predictive score variable[37]. In this case, the predictive score is the hazard, as it estimates the risk a system is in at a given time. The C-Index compares the relationship between the actual lifetime and the hazard of a particular model. Its computation is also relatively straight forward and is capable of being extended to include censorship.

In the case for non-censored observations, the following applies: Let (t_i, h_i) be the observed time and hazard of an individual system, i, in a collection of n systems.

Probability of concordance is defined as:

$$\mathbb{P}_c = \Pr(t_i < t_j \text{ AND } h_i < h_j \text{ OR } t_i > t_j \text{ AND } h_i > h_j)$$

Probability of discordance is defined as:

$$\mathbb{P}_d = \Pr(t_i < t_j \text{ AND } h_i > h_j \text{ OR } t_i > t_j \text{ AND } h_i < h_j)$$

Then, generally the C-Index is:

$$C_{t,h} = \frac{\mathbb{P}_c}{\mathbb{P}_c + \mathbb{P}_d}$$

As can be seen from the above, the C-Index is simply the fraction of concordant pairs to all pairs. Put another way, it is the proportion of times the hazard correctly orders the lifetime of a system. Of course, the above is only applicable if all lifetimes are observed and, as noted earlier, this is rarely the case. However, to extend the C-Index to censored observations a more imperative version of the above is required.

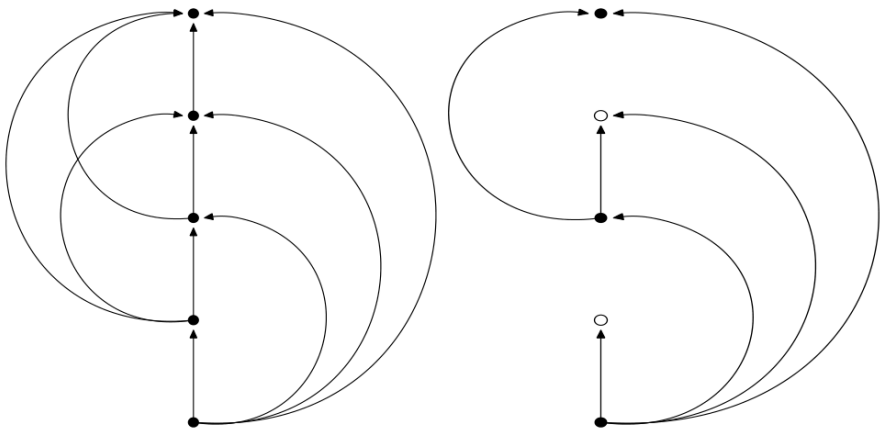

Figure 3: Order Graph of C-Index Without and With Censoring

To introduce the effect of censored observations, it is easier to think of time-to-event data as an ordered graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and as displayed in Figure 3[67]. Each observation is a triple, (t_i, h_i, d_i), with the observed time, hazard and censoring indicator for any individual system. The set of vertices $\mathcal{V}$ represent all individual triples in the data set. Each vertex, $\mathcal{V}_\rangle$, is indicated to be either an event or censored using the censoring indicator, d_i. Edges, $\mathcal{E}_{ij}$ between the vertices of the graph are only drawn such that $t_i < t_j$ and no edges can come from a censored observation.

In such a context, the C-Index is:

$$C(t, h, d, \mathcal{G}) = \frac{1}{|\mathcal{E}|} \sum_{\mathcal{E}_{ij}} I(h_i < h_j)$$

With $I(\cdot)$ being the indicator function, which is one if $h_i < h_j$, and zero otherwise. The $|\mathcal{E}|$ is the number of edges in the graph.

It is important to take a moment and understand what this construction implies. Censored observations contribute to the C-Index only in one direction. Specifically, if and only if, an uncensored failure time is smaller than a censored survival time. This makes intuitive sense. If a pair of observations are given, i, j, where $t_i < t_j$, then if the first observation, i, is an event it can still be compared to the censored observation later in time. If the score of the first observation is larger, then a concordant pair results, if not, the result is a discordant pair. However, if the first observation is censored, nothing can be said about the pair. This is because, by definition, a censored observation will have a lifetime greater than that of an event. So there is nothing to determine whether that censored observation will continue to survive until it has a lifetime less than or greater than the event observation.

The C-Index can have this one-directionality made more explicit in the following way[67]:

$$C(t, h, d, \mathcal{G}) = \frac{1}{|\mathcal{E}|} \sum_{t_i | d_i = 1} \sum_{t_i < t_j} I(h_i < h_j)$$

The C-Index is a generalization of the Area Under Curve (AUC) or the Receiver Operating Conditions (ROC) plot, both of which assess the predictive performance of a binary classifiers[12][70]. Clearly in this context, the goal is not binary classification, rather the correct ordering of events. However, a similar interpretation result occurs as in other contexts. The C-Index has a range of $[0.5, 1]$. At one extreme a C-Index is one, which implies the model produces a perfect prediction, such that the rankings of each hazards are concordant with each lifetime across all observations. At the other extreme, 0.5, the model has a probability equivalent to flipping a coin.

The C-Index does have its drawbacks. While the C-Index has objective value as a metric for establishing the predictive accuracy of a model, it is less suitable for selecting which set of covariates produce the best model. This is because it is relatively insensitive to the inclusion of covariates[12]. Different sets of covariates are unlikely to result in substantial changes in the C-Index's numerical value. The lack of substantial variation can make the task of model selection somewhat more difficult. Thus, it is important to complement the C-Index with more traditional methods of assessing the goodness-of-fit of a model. Highly recommended are the use of methods that derive out-of-sample deviance based on Information Criteria.

In the Bayesian context, the Widely Applicable Information Criterion (WAIC) is a novel metric that provides point-wise average out-of-sample deviance[75]. The WAIC is an improvement over the Deviance Information Criterion (DIC) which is commonly used in Bayesian analysis. It addresses several of the shortcomings that are associated with this metric, mainly stemming from the DIC being computed around a point estimate.

Briefly, the WAIC approximates the expected log posterior predictive density for a new dataset[74].

$$\text{elpd} = \sum_{i=1}^{n} \int p_t(\tilde{y}_i) \ln p(\tilde{y}|y) d\tilde{y}_i$$

Where $p(\tilde{y}|y) = \int p(\tilde{y}|\theta)p(\theta|y)d\theta$ is the posterior predictive distribution and $p_t(\tilde{y}_i)$ is the distribution of the true data generating process. Clearly, the true data generating process is not observed. However, it can be approximated the following way:

$$\hat{\text{elpd}} = \hat{\text{lpd}} - \hat{p}$$

Where $\hat{\text{lpd}}$ is the log pointwise predictive density and the $\hat{p}$ is the number of effective parameters in the model. The log pointwise predictive density is simply the probability of observing each one of the data values given the set of parameter values used in the model.

$$\text{lpd} = \sum_{i=1}^{n} \ln \int p(y_i|\theta)p(\theta|y)d\theta \qquad \hat{\text{lpd}} = \sum_{i=1}^{n} \ln \left(\frac{1}{S} \sum_{i=1}^{S} p(y_i|\theta^s) \right)$$

Where θ^s is the vector of completed posterior simulations.

The effective number of parameters is approximated using the sample variance of the log likelihood of posterior distribution.

$$\hat{p} = \sum_{i=1}^{n} \text{VAR}[\ln p(y_i|\theta^s)]$$

Where $Var[\cdot]$ is the sample variance.

This entire process is easily implemented in Stan as it can be directly computed in the `generated quantities` block[73]. As with all information criterion metrics, the lower the value the less deviation there is from the true data generating distribution.

The C-Index and WAIC evaluate performance but through different means. Both assess the generalization of the model to new data. But both have substantially different practical interpretations. Yet, if the model is stable, both should result in the same conclusion being reached about accuracy and discrimination. Yet, if an error in model design or fit has occurred, their lack of harmony should be a signal to reevaluate the current model.

Stan

Stan is the most recent addition to the toolkit for fitting Bayesian models. Like its predecessors, BUGS and JAGS, Stan seeks to abstract the task of fitting complex models with a general programmatic process. Stan makes use of Hamiltonian Monte Carlo (HMC) for the sampling of continuous parameters from a model. The existence of Stan stems from the operational difficulties of its predecessors in dealing with multilevel generalized linear modeling contexts. These software packages have requirements, such as conjugate priors or log-concave posterior densities, for models. This made anything outside of those rigid constraints incredibly inefficient when sampling was performed. In the current context, the introduction of complex shared Frailties can create similar problems with these existing modeling tools.

A Stan program defines a statistical model through a conditional probability function, $\Pr(\theta|y,x)$, where θ is a sequence of unknown modeled values while y is composed of modeled known values, and x of unmodeled covariates and constants[66]. Stan is an imperative probabilistic programming language. This implies that the language requires the user to tell it how to do something more than just declare that it should be done. Precise instructions have to be written verbatim to achieve the desired modeling outcome. Yet, as the previous chapter demonstrated, models are constructed by combining relatively simple building blocks. This permits models of any complexity to be defined and built upon.

```
functions {
  // ... function declarations and definitions ...
}
data {
  // ... declarations ...
}
transformed data {
  // ... declarations ... statements ...
}
parameters {
  // ... declarations ...
}
transformed parameters {
  // ... declarations ... statements ...
}
model {
  // ... declarations ... statements ...
}
generated quantities {
  // ... declarations ... statements ...
}
```

Figure 4: Available Stan Blocks

A Stan program consists of variable declarations and statements, divided into a series of blocks written in a C-like syntax. This compartmentalizes operations, as variables can be declared in the block within which they are used. Generally, three blocks are used to define a statistical model. These are the data, parameters and model blocks. Figure 4 demonstrates all the available blocks and their required order.

The data block declares the data required for the model. The parameter block declares the model parameters, or the unobserved random variables being sampled by the model. The model block defines the log probability function used to fit the model.

The data and parameter blocks both handle declaration of variables. In Stan, random variables are handled differently depending on whether or not they are observed. Observed random variables are declared as data while unobserved random variables are declared as parameters. Unobserved random variables can be sampled from or inserted in subsequent blocks. This is especially useful in the computation of the log probability function.

To facilitate processing two transformation blocks and generated quantities block are available. The transformation blocks allow for data and parameters to be altered and saved in the process of executing a Stan program. This provides additional flexibility in model building. The generated quantities block allows for the creation of values of importance, such as summary statistics. The block is run at the end of each sampling step, this permits for the tracking of intermediary values as the model is fit.

Once a Stan program is defined, it is compiled into `C++` code before being executed. The program begins by validating the known values of y and x and checking their types and constraints. It then generates a sequence of non-independent identically distributed parameter values, which have a marginal distribution of $\Pr(\theta|y,x)$. This translation into `C++` code greatly improves the speed and portability of Stan-built models. However, it also results in requirements that do not arise in earlier Bayesian samplers. Notably, Stan is statically typed and requires that all variables have their constraints explicitly defined, if they exist. This is partially to enforce explicitness among model designers, but it is also to prevent modeling errors from creating problematic outputs. For example, a Weibull baseline hazard requires that both the shape and scale parameters be strictly positive otherwise the function is not defined. There is no reason for any sampler to make use of negative parameter values when sampling from these distributions. Strict type setting with constraints ensures that these types of errors are caught as soon as they occur rather than potentially disturbing final results.

Once the sampler runs until convergence, Stan writes the output values of each parameter to disk in a `csv` file. This permits sampling to be done multiple times with the values of each iteration updating the output. This is particularly useful for large models that may require a greater amount of computer time for each iteration.

Stan comes with a great many of built-in operations and functions. The basic operations are the same as one expects from any programming language, logical and arithmetic operations, matrix and array manipulation, type conversions as well as built-in functions for handling mathematical operations like solving ordinary differential equations. The language also includes functions that encode most statistical distributions including those that are found in the previous chapter. Furthermore, sampling statements are used that vectorize the sampling from known distributions improving the speed of execution.

While these built-ins are useful for the vast majority of tasks, time-to-event models pose a problem for 'vanilla' Stan. Much of the difficulty in building time-to-event models in Stan stems from the presence of censoring and truncation in the data. While Stan does have some support for dealing with censoring and truncation, the nature of this support is fairly inflexible, especially when dealing with more complex models, such as those with time-dependent variables or Frailties. The recent introduction a function block in the language has largely enabled the fitting of time-to-event models in Stan. As the name suggests, the function block allows for any arbitrary function to be declared. This

includes the ability to explicitly define a likelihood function. This mechanism permits the development of time-to-event models as the effect of censoring can be written explicitly into the models.

Building The Model

This section brings all the previous insights together and provides a step-by-step process to fitting and evaluating time-to-event models. It begins with the construction and output of a simple Multiplicative Hazard Model, it then extends to this model to accommodate shared Gamma Frailties. The emphasis in this section is on practical exposition, over theory. As a result, some implementation details may be omitted for the sake of brevity.

It is important to start simple with model construction. A good first step is to fit a basic Multiplicative Hazards Model, in this case with a Weibull baseline hazard.

```
functions {
  real log_h_t(real lifetime, real alp, real gam);
  real H_t(real lifetime, real alp, real gam);

  real log_h_t(real lifetime, real alp, real gam){
    return log( (alp/gam) ) + (alp - 1) * log( (lifetime / gam) );
  }

  real H_t(real lifetime, real alp, real gam){
    return ( (lifetime / gam) )^alp ;
  }

  real surv_dens_log(vector cens_lifetime, real alp, real gam, real
   lin_pred){
    real lifetime;
    real d_i;

    lifetime <- cens_lifetime[1];
    d_i      <- cens_lifetime[2];

    return d_i * (log_h_t(lifetime, alp, gam) + lin_pred) - H_t(lifetime
   , alp, gam) * exp(lin_pred);
  }
}
```

Figure 5: Function Block for Multiplicative Hazard Model

Stan permits the inclusion of arbitrary functions. These functions can be used in any manner, but in this context they are primarily useful for evaluating the likelihood of the model. To be included, they must be placed in a `function` block at the beginning of the model code. Figure 5 contains code block for the functions needed to define a Multiplicative Hazard Model in Stan. The conditional cumulative hazard, $H(t|x)$, and the conditional log hazard, $\ln(h(t|x))$, are defined in the `H_t()` and `log_h_t()` functions, respectively. The model's log likelihood, $\ell(\theta|Y, d)$ is defined in the `surv_dens_log()` function. This last function that will be minimized in sampling steps of the fitting process.

The process of explicitly writing likelihoods exposes one to the inner operation of a model. This fosters a deeper connection between the mathematics and programming.

Furthermore, such a process enables one to extend the model far beyond what is covered in this text.

It should be noted, that this type of functional definitions are not commonly required to use Stan, as it comes with a large library of built in functions[66]. However, Stan's handling of censored data is currently problematic for most time-to-event models making such functions mandatory. To make the effect of censoring explicit, the likelihood function takes a vector as input `cens_lifetime` that contains the double, time and censoring indicator, (Y_i, d_i), needed to define an observed lifetime.

```
        data {
          int<lower=0>  N;
          vector[2]     lifetime[N];
          real          x_err1[N];
          real          x_err2[N];
          real          x_repair[N];
          real          x_kWh[N];
          real          park_weather[N];
        }
```

Figure 6: Data Block for Multiplicative Hazard Model

Stan is statically typed and forces precision in definitions. As a result, every function and variable must have its type explicitly stated before hand. These are provided in Figure 6. Most common data types are available, including ones for dealing with vector and matrix operations. As these functions deal with continuous variables, `real` is the common type. Due to the language's imperative nature, order is not arbitrary and all variables and functions must be defined prior to being used.

```
parameters {
  real<lower=0> alp;
  real<lower=0> gam;
  real          beta_err1;
  real          beta_err2;
  real          beta_repair;
  real          beta_kWh;
  real          beta_weather;
}
```

Figure 7: Parameters Block for Multiplicative Hazard Model

To allow for the sampling of the likelihood, input data must be defined. These are the observed values. Data can be input any number of different ways. Stan is self-sufficient and it can be accessed through a variety of programming languages, like R, Python, Julia, etc. Each one of these languages have the ability to feed data into Stan. In the R language, this is done through a named `list()` The names within the list should correspond to the variable names found in the `data` block in Figure 6. The number of observations, the values of those observations and the values of any covariates must be defined here. Note, that the type and dimensions of each element of data must be declared here. It is good practice to center and scale data prior to input. Even if a covariate is a count, like the number of days since repairs or a count of error codes, it is advisable to scale and convert it to real values. This greatly improves the performance of Stan, as HMC is optimized to

operate on continuous functions. If this is not done a much slower discrete optimization process is used. Of course, caution is advised as it is possible that the data one wishes to import cannot take on real values.

The parameters of the model also must be defined. This is done in Figure 7. In this case, this includes the parameters for the Weibull distribution, `alp` and `gam`, as well as the linear coefficients, `beta_`, associated with the model covariates. Notice the `real<lower=0>` in the parameter definitions. These are constraints which are applied to the `alp` and `gam` parameters to ensure that their support is only over the positive real numbers. In this model, failure to do so will result in errors produced by negative probabilities, causing proposal moves to be discarded. This may result in poor convergence and biased parameter estimates. Even in situations where where the sampler does not explicitly warn about negative probabilities, it is best practice to apply constraints if one is aware of them.

```
model {
  for(i in 1:N){
    lifetime[i] ~ surv_dens(alp, gam, beta_err1 * x_err1[i] + beta_err2
   * x_err2[i] + beta_repair * x_repair[i] + beta_kWh * x_kWh[i] +
   beta_weather * park_weather[i]);
  }

  alp ~ lognormal(0, 1.5);
  gam ~ lognormal(0, 1.5);

  beta_err1 ~ normal(0, 1);
  beta_err2 ~ normal(0, 10);
  beta_repair ~ normal(0, 1);
  beta_kWh ~ normal(0,10);
  beta_weather ~ normal(0, 1);
}
```

Figure 8: Model Block for Multiplicative Hazard Model

The model block encapsulates the high-level configuration of the model and defines the sampling process. Figure 8, defines both the priors as well as the method of sampling. The sampling is declared through a sampling statement. The `for()` loop iterates over each element in the data and declares that each observation is distributed, $Y \sim \mathcal{L}(\theta)$, given a likelihood. In this case, as a log-likelihood was already defined in the `function` block it is possible to make use of it. Note that the function being called is `surv_dens` not `surv_dens_log` as was defined in the `function` block. This is because each observation is defined according to the likelihood, even if the model is fit using a log-likelihood. Stan understands the `_log` suffix and acts accordingly. This is meant to ensure that the focus is retained on the characteristics of the model rather than its fitting process.

Below the sampling statement are the priors associated with the parameters. Priors do not have to be explicitly defined in Stan. If they are not, Stan will use a continuous uniform prior. However, these are highly inefficient especially if no constraints were declared for a particular parameter. The consequences of this omission can be poor convergence and biased estimates. This is especially true of the scale parameter of the Weibull distribution, `gam`, whose true value is often far from zero. Weakly regularizing priors centered near zero are advisable, especially if all the variables were scaled beforehand. If there is some

existing knowledge about the expected lifetime of a system, a prior can also be used as a means by which to introduce this expertise into the model.

```
generated quantities {
  vector[N] pred_log_h_t;
  for(i in 1:N){
    pred_log_h_t[i] <- log_h_t(lifetime[i,2], alp, gam, beta_err1 *
   x_err1[i] + beta_err2 * x_err2[i] + beta_repair * x_repair[i] +
   beta_kWh * x_kWh[i] + beta_weather * park_weather[i]);
  }
}
```

Figure 9: Generated Quantities Block for Multiplicative Hazard Model

Lastly, it is possible to directly output the posterior predictive distribution for each inverter using the `generated quantities` block. This is shown in Figure 9. To achieve this, a simple iteration across each observation using the final parameters is done. These values can then be extracted into R and their predictive accuracy assessed. Most importantly, it is possible to extract the variances on each prediction, allowing this to be taken into account when planning maintenance scheduling. Generally, Stan provides an entire class of functions for numerous distributions with the suffix, `_rng` that allow for the generation of these quantities. However, as a custom likelihood is used, so too must a custom range function.

```
               mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alp            2.96       0 0.13  2.72  2.87  2.96  3.04  3.21  4797    1
gam            5.10       0 0.10  4.91  5.03  5.10  5.17  5.30 11654    1
beta_err1      0.21       0 0.06  0.10  0.17  0.21  0.25  0.33 10551    1
beta_err2      2.99       0 0.13  2.73  2.90  2.98  3.07  3.24  4844    1
beta_repair   -0.23       0 0.05 -0.34 -0.26 -0.23 -0.19 -0.12 12435    1
beta_kWh      -3.92       0 0.17 -4.26 -4.04 -3.92 -3.81 -3.60  4897    1
beta_weather   0.25       0 0.06  0.14  0.21  0.25  0.28  0.36  9761    1
```

Figure 10: Output of Multiplicative Hazard Model

Once the model is fit, the parameters can be easily printed out or plotted, with the `print()` and `plot()` functions respective. The `print()` output can be seen in Figure 10. It includes what is generally expected from a linear model output, the mean values of the posterior, standard errors, the distributions standard deviations and quantiles. Additionally two metrics for diagnosing convergence issues are given. The `n_eff` is a crude measure of effective sample size. As the MCMC process produces autocorrelated samples, the effective samples are an estimate of the number of independent samples in the process, this number should be high relative to the number of samples given. Very low numbers can be a sign of bias in the outcome. The `Rhat` is the Gelman-Rubin statistic, which can roughly be thought of as a estimate of the convergence of the Markov chains to the target distribution[24]. If convergence occurs, `Rhat` approaches 1 from above. As such values greater than one can suggest that the model has not yet converged and the samples should not be trusted. Alternatively, convergence can be checked visually through the use of a traceplot. Figure 11 illustrates the traceplot for the `alp` and `gam` parameters. Ideally, there should be no visible pattern in the traceplot and it should be centered around the mean of the parameter value.

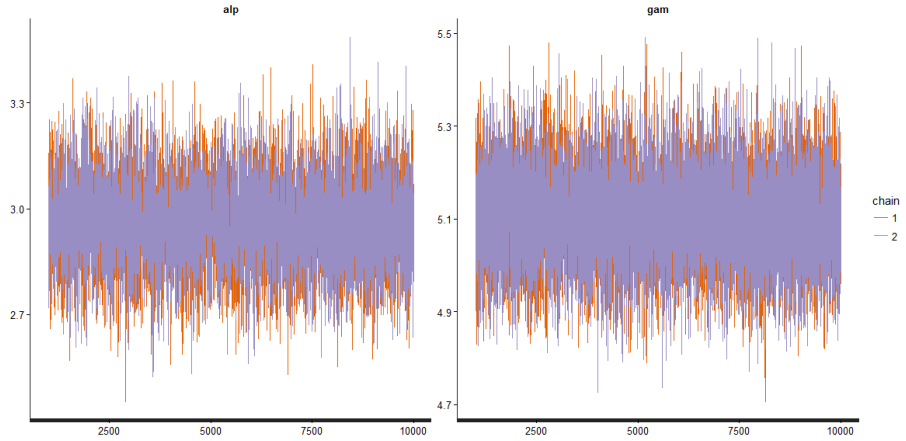

Figure 11: Traceplot of `alp` and `gam` Parameters

```
generated quantities {
  vector[N] log_lik;
  for (i in 1:N){
    log_lik[i] <- surv_dens_log(lifetime[i], alp, gam, beta_err1 *
    x_err1[i] + beta_err2 * x_err2[i] + beta_repair * x_repair[i] +
    beta_kWh * x_kWh[i] + beta_weather * park_weather[i]);
  }
}
```

Figure 12: Log Likelihood in Generated Quantities Block

Once it is clear that the model has converged to a stable solution. Fit and prediction accuracy can be assessed using WAIC and the C-Index. To generate the WAIC metric, a point-wise log likelihood, `log_lik`, is required. Figure 12 demonstrates how to retrieve this value. The `generated quantities` block allows for these likelihoods to be extracted upon completion of sampling. The functions `WAIC()` from the `rethinking` package can be used to compute the WAIC value for the model under the assumption that the `log_lik` vector is included in the output.

To compute the C-Index, the hazard function, the prognostic measure, must first be generated. This is done by sampling the parameters from posterior. This has the added benefit of retaining the uncertainty associated with each parameter as well as the hazard function. As stated earlier, the C-Index should be computed on a hold-out sample to prevent overfitting. The simulated data used for this exploration, provides the added ability to estimate the C-Index on the uncensored lifetimes. This completes the process.

Unobserved heterogeneity or clustered observations can be implemented in several ways in Stan, with roughly the same results. However, the strategy for what is being optimized varies.

First, it is possible to repeat the mechanism with which the Multiplicative Hazard was fit, by explicitly writing out the log likelihood into the `function` block. Recall, that the Frailty, Z, is not directly modeled, as it is constrained to have an expectation of one, $\mathrm{E}[Z] = 1$, by design. Rather the unconditional likelihood is used with the Frailty parameter integrated out. Figure 13 illustrates the necessary changes required for the density function.

```
real surv_dens_log(vector cens_lifetime, real alp, real gam, real sig,
  real lin_pred){
  real lifetime;
  real d_i;
  real sig_i;

  sig_i    <- sig;
  lifetime <- cens_lifetime[1];
  d_i      <- cens_lifetime[2];

  return d_i * log(sig^2) + log(tgamma((1/sig^2) + d_i)) - log(tgamma(
  (1/sig^2) )) -
          ((1/sig^2) + d_i) * log(1 + sig^2 * H_t(lifetime, alp, gam)
 * exp(lin_pred)) +
          d_i * (lin_pred + log_h_t(lifetime, alp, gam));
}
```

Figure 13: Gamma Frailty Likelihood in Function Block

Beyond the density function, there is also a requirement to resolve the sums across the clusters. To do this a grouping variable must be made during the model fitting process. This allows for the estimation of σ^2 for each level of the group. In this example, the each park is given its own estimate of the Frailty variance. The estimation process is done as before, but a different value is given for each of the five parks in the input data.

To do this in Stan must be made aware that the parameter `sig` will now be a vector of real numbers. It needs to know the size of the vector. Then, the fitting process must take into account which position in the vector of `sig` sampling is being done. Thus, the following additions are required to complete the Stan program:

```
data {
  int<lower=1>  park[N];
  int<lower=1>  park_N;
}

parameters {
  real<lower=0> sig[park_N];
}

model {
  for(i in 1:N){
      lifetime[i] ~ surv_dens(alp, gam, sig[park[i]], beta_err1 * x_err1
   [i] + beta_err2 * x_err2[i] + beta_repair * x_repair[i] + beta_kWh *
    x_kWh[i] + beta_weather * park_weather[i]);
  }
  for(j in 1:park_N){
    sig[j] ~ lognormal(0, 1.5);
  }

}
```

Figure 14: Park Grouping in Stan Blocks

Figure 14 demonstrates the required changes. The `data` block introduces a grouping variable, `park` which is the length of the number of observations, `N`. It identifies park mem-

bership of any single observation with an integer value. The `park_N` variable is introduced to provide a count of the total number of parks in the whole dataset. This is useful in the `parameters` block for defining the dimensions of the `sig` vector that stores the variances of the Frailty parameter. Finally, in the `model` block, the same sampling statement is made as before, however, now the additional indexing of the `sig` parameter is introduced. This is done through the use of an index on the `park` variable, which has a range between one and five. As such, when the sampling statement is run, a value at each park is generated within, `sig[park[i]]`.

```
real surv_dens_log(vector cens_lifetime, real alp, real gam, real
 lin_pred, real Z){
  real lifetime;
  real d_i;

  lifetime <- cens_lifetime[1];
  d_i      <- cens_lifetime[2];

  return d_i * (log_h_t(lifetime, alp, gam) + lin_pred + Z) - (H_t(
 lifetime, alp, gam) * exp(lin_pred) * exp(Z));
}
```

Figure 15: Fraily Model with Direct Estimation of Z parameter

An alternative method can be used to fit the model in a substantially more flexible and extensible form. This allows for the direct sampling of the Z parameter rather than integrating it out through a constraint. This greatly simplifies the model. Figure 15 provides the likelihood function, `surv_dens_log`, needed to fit the model. The other alterations required to the code are the same as in in the previous case, a vector of Z parameters must be defined in the `parameter` block and added to the end of the likelihood function call in the `model` block. As before, a park grouping should be used like `Z[park[i]] to ensure the sampling is done for each park. As can be seen from Figure~\ref{z_lik`, the introduction of the Z parameter simply requires the inclusion of a single term to the hazard and cumulative hazard of the likelihood. This likelihood is the original simple likelihood of the Multiplicative Hazard Model.

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
alp	2.92	0.00	0.19	2.55	2.79	2.92	3.06	3.30	6818	1
gam	6.79	0.01	0.83	5.57	6.22	6.65	7.22	8.77	5811	1
Z[1]	0.57	0.01	0.47	0.02	0.21	0.46	0.80	1.78	6111	1
Z[2]	0.48	0.01	0.38	0.02	0.19	0.39	0.67	1.46	5266	1
Z[3]	0.97	0.01	0.37	0.31	0.72	0.95	1.20	1.79	5253	1
Z[4]	0.41	0.00	0.33	0.01	0.16	0.34	0.58	1.23	6143	1
Z[5]	0.48	0.00	0.38	0.02	0.19	0.39	0.67	1.45	5938	1
beta_err1	0.01	0.00	0.10	-0.18	-0.05	0.01	0.08	0.20	12254	1
beta_err2	2.78	0.00	0.18	2.44	2.66	2.78	2.90	3.13	6978	1
beta_repair	-0.16	0.00	0.10	-0.36	-0.23	-0.16	-0.10	0.03	13082	1
beta_kWh	-3.87	0.00	0.25	-4.36	-4.03	-3.87	-3.70	-3.39	6733	1
beta_weather	0.19	0.00	0.27	-0.35	0.02	0.18	0.35	0.77	6185	1

Figure 16: Output of Frailty Model

Figure 16 provides the standard output for the Frailty model. As before, it includes what is expected, the mean values of the posterior, standard errors, the distributions

standard deviations and quantiles for each parameter. Additionally, the Z parameters are presented which express the hazard at each park. The larger than average value of `Z[3]` illustrates that there is some systemic problem at that park that maintenance staff may wish to address. While the structure of the model, not the output, improves the prediction, the ability to understand the rationale for those predictions is a valuable tool for enabling better decision making about how to allocate resources.

This completes the shared Frailty extension. The remainder of the process is identical to that which was performed above. Upon completion the basic sanity check of statistical modeling must be performed. Assumptions regarding the adequacy of the baseline hazard and the convergence of the model should be checked in the same manner as described before. The C-Index and WAIC should be used to check the predictive performance as well as to ensure that the model is not overfitting. Finally, a list of the most at-risk inverters can be output.

One of the wondrous benefits of working with Stan, is that the models constructed above can be extended indefinitely with simple additions. This can be done in the same manner as presented previously through the addition of Frailties to deal with unobserved heterogeneity. Numerous potential extensions are feasible, only an understanding of their structural affect must be integrated. For example, it is possible to stratify the baseline hazard allowing a different shape and scale for each group. It is also possible to expand the Frailty beyond the current independence assumption and provide a correlation structure between terms in the model. It is even possible to directly model time-varying effects, to explicitly cope with phenomena whose relationship changes as the system ages. The possibilities need only to be justified, tested and checked to ensure they provide improvements in out of sample performance. The extension potential is limitless.

Extended Reading

For an excellent guide to feature engineering of time-dependent processes, the reader is directed to Microsoft Playbook by Uz[72]. Inverse transform sampling in time-to-event models is covered by Bender, Augustin and Blettner[5], is extended to Frailties by Romdhane and Belkacem[61] and into time-dependent covariates by Austin[4] and Hendry[30]. For more information on the Stan probabilistic programming language, the Stan development team has a exhaustive manual in place[66]. For development of measures to assess the accuracy of predictive models or any models, the classic text by Harrell is highly recommended[29]. For more information about WAIC, Vehtari, Gelman and Garby cover the derivation and properties[74]. For an overview of how to implement models in Stan, the massive Stan Reference Manual is likely without equal[66]. For slightly less overwhelming information about Stan, the documentation page on the project website contains a wide assortment of tutorials, videos and softer introductions to the software[65].

Conclusion

The preceding chapters addressed the development of predictive maintenance process in informationally sparse systems. They sought to fill the void in the current literature on predictive maintenance in domains with limited access to operational data. To address this vacuum, the lens of Bayesian time-to-event modeling, a statistical approach used to estimate the remaining lifetime of an object, was used. The text was split into three broad sections, the photovoltaic domain, the mathematical construction of time-to-event models and the analytical process.

An example domain, which serves as a business case, was presented. The solar industry, with its heavy reliance on dwindling government subsidies and firm requirement to reduce costs serves as fertile ground for a predictive maintenance process. It is a domain where any incremental improvement in profitability can provide a substantial competitive advantage. The photovoltaic inverter, embedded in an industrial-scale solar power plant is an archetype of a sparse information system. A system with limited intelligence, where, at least for the time being, the addition of sensors to facilitate better condition-based monitoring is unlikely.

Predictive maintenance, its underlying theory, use-cases and requirements were also discussed. The purpose of predictive maintenance is ultimately to provide companies with the ability to adequately plan for servicing and schedule downtime. To empower industry to be in control of their operational process rather than being at the mercy of environmental externalities. However, to achieve such a goal, the ability to implement corrective actions as well as gather and store reliable historical information related to system reliability is essential. Required are detailed maintenance logs, failure history, system attributes and any available conditional monitoring data. Domain subject matter expertise should be exploited to ensure that these records are relevant, timely and complete.

These demands were then examined within the context of the solar industry. A detailed list of requirements for enabling a predictive maintenance process on this system were given, as well as many of the difficulties and barriers that arise from the domain. This provided an awareness of the state of the industry as well as what pitfalls may be encountered in an effort to develop a predictive maintenance process.

Then, the focus shifted to the details required for the construction of Bayesian time-to-event models. Specifically focusing on the mathematical details that are necessary prerequisites for the models' development. The process was begun from first principles, with the basic functions that define an object's lifetime being presented. Special care was taken to provide intuition into objects unique to these type of models, like hazard, survival and cumulative hazard functions. Additionally, relationships between the functions which allow for transformations between any of the functions that define lifetimes was given.

A list of common lifetime distributions was provided. This began with a description

of a constant hazard from which the exponential distribution arises. The exponential was demonstrated to the be the source of the most widely used parametric lifetime distributions, the Weibull, Gompertz and Gamma. Each mathematical construction of the hazard was provided along with an understanding of how each distribution can be said to aggregate different types of failure events.

Censoring and truncation was then explored, which are defining elements of time-to-event analysis. Due to the nature of the problem, full lifetimes are rarely observed and some degree of missing data is found in any analysis. The contrast was drawn between known-unknowns, censoring, and unknown-unknowns, truncation, in terms of their respective effects on time-to-event models. A mathematical construction was then provided for how to define censoring within the model through the use of a dummy variable.

The introduction of covariates into lifetime models was explored. Covariates serve as the means by which the model integrates environmental conditions with estimates of lifetimes. The Multiplicative Hazards Model was introduced. This model, as the name suggests, creates a multiplicative relationship between the hazard functions and a set of linear predictors.

Frailties were then introduced as a method for accounting for unobserved heterogeneity or correlation between observations. The previous models all assumed independent and identically distributed observations and the Frailty allowed for these assumptions to be relaxed. Specifically, the shared Gamma Frailty was introduced into the Multiplicative Hazard's Model. Its construction was provides as well as the restrictions needed to ensure model identifiability.

Model likelihoods, the functions used to select the appropriate abstraction to the observed data, were then discussed. Each model discussed had its likelihood presented in detail, as it would be relevant to the implementation process later on. Two alternative methods of introducing Frailties were given.

The Bayesian model estimation process was explored. The usefulness of being able to avoid optimization difficulties associated with traditional fitting processes were discussed. This included previous Bayesian tools for model fitting. Then a brief explanation of Hamiltonian Monte Carlo was provided, that outlined how the process generates samples from the posterior based on an innovative means of selecting proposal distributions.

The last chapter covered the implementation of a predictive maintenance process. It addressed the details that, more often then not, are disregarded in discussions about analytical processes.

It formally stated the desired outcome for the model construction. Specifically, a score function that ranks the order of inverter failure. This does away with the need for an exact determination of failure time and instead recasts the problem as a rank-order prediction. Such an output provides sufficient information for the scheduling of corrective maintenance activities.

Feature engineering, the dark art of building model covariates was explored. Its aim is to produce variables that accurately describe a photovoltaic inverter's health at any given moment. A task, if done correctly, is the difference between an effective and ineffective analytical process. Two types of knowledge were shown to be essential to feature engineering, domain-specific and model-specific subject matter expertise. The domain-specific subject matter expertise encompassed an understanding of the system under observation and the various phenomena that contribute to its failure. Model-specific subject matter

expertise refers to the an understanding of the details and limitations of the mathematical construction used to predict lifetimes. Specifically an understanding of what types of covariates will have the greatest amount of traction in which models. Experimentation was also discussed and the role it plays in the iterative construction of models.

Data management and simulation were discussed. The requirement for attribute-value data for the modeling process was given. This requirement was expressed both mathematically and through an example table. Then, the structure for the simulated data sets was provided with each respective covariate explained. The simulation process used to generate this data was provided. It was developed from first principles to allow for the simulation of data from progressively more complex data structures.

An overview of the probabilistic programming language, Stan was then provided. It was distinguished from its predecessors in terms of being able to implement models that created difficulties for other software. The underlying logic of the imperative language was provided, with an description of programming blocks needed to define a statistical model. Some caveats and features were also discussed which differ from other statistical software including static typing, constraints and the ability to save iterations to file.

Evaluating predictive performance was then examined. The challenges of underfitting and overfitting were presented as the reasons for the need for adequate performance metrics. The difficulties with using the most common methods for evaluating models was discussed. The sequencing of time-to-event data along with great imbalances caused by censoring were identified as the primary sources of the lack of effect of these traditional methods. The C-Index was presented as a well-defined method for evaluating predictive performance of time-to-event models. It was developed in mathematical detail and extended to include censored observations through the realization of its one-way effect.

The final section brought together all the pieces and built a series of models in Stan. It demonstrated how the modeling process can be performed in an iterative fashion. Building on first principles and directly incorporating functions into Stan. First, with the simplest Multiplicative Hazard Models. This allowed for an exposition of how the various pieces of Stan code can be used to express the mathematical constructions described in the previous chapter. It also allowed for a brief discussion on model validation steps that should be taken after fitting any model. This entire process was then extended to include Frailty models.

Predictive maintenance is a rapidly evolving field. However, it is one that will undoubtedly continue to progress toward a greater utilization of a wider breadth of data, leaving informationally sparse systems under-represented. These systems will either become informationally rich as the industries mature and data capture becomes easier, or they will continue to make use of statistical methodologies that are able to optimally make use of the limited amount of data they generate.

Link to Code

The code for the models can be found at:
https://github.com/mrdevlar/MscPM

List of Figures

Bibliography

[1] O. O. Aalen, O. Borgan, and H. K. Gjessing. *Survival and Event History Analysis: A Process Point of View*. Springer, New York, 2008. ISBN 9780387202877. 550 pp.

[2] P. K. Andersen, O. Borgan, D. R. Gill, and N. Keiding. Statistical Models Based on Counting Processe, 1992. ISBN 0387945199.

[3] J. Anguita, M. Ahmad, S. Haq, and J. Allam. Ultra-broadband light trapping using nanotextured decoupled graphene multilayers. *Science*, 2(2), 2016. URL `http://advances.sciencemag.org/content/2/2/e1501238.abstract`.

[4] P. C. Austin. Generating survival times to simulate Cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29):3946–58, dec 2012. ISSN 1097-0258. URL `http://www.ncbi.nlm.nih.gov/pubmed/22763916` `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3546387`.

[5] R. Bender, T. Augustin, and M. Blettner. Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, 24(11):1713–23, jun 2005. ISSN 0277-6715. URL `http://www.ncbi.nlm.nih.gov/pubmed/15724232`.

[6] J. Blitzstein and J. Hwang. *Introduction to probability*. CRC Press, Boca Raton, 2014.

[7] H. C. Boshuizen and E. J. Feskens. Fitting additive Poisson models. *Epidemiologic perspectives & innovations : EP+I*, 7:4, 2010. ISSN 1742-5573. URL `http://www.ncbi.nlm.nih.gov/pubmed/20646285` `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2914659`.

[8] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, and B. Goodrich. Stan: A probabilistic programming language. *Journal of Statistical Software*, VV(II):43, 2016. URL `http://www.uvm.edu/~bbeckage/Teaching/DataAnalysis/Manuals/stan-resubmit-JSS1293.pdf`.

[9] D. Carrington. Carbon bubble will plunge the world into another financial crisis - report, apr 2013. URL `https://www.theguardian.com/environment/2013/apr/19/carbon-bubble-financial-crash-crisis`.

[10] C. C. Ciang, J.-R. Lee, and H.-J. Bang. Structural health monitoring for a wind turbine system: a review of damage detection methods. *Measurement Science and Technology*, 19(12), dec 2008. ISSN 0957-0233. URL `http://stacks.iop.org/0957-0233/19/i=12/a=122001?key=crossref.1c9821fda788dd43848a970cbd8da0aa`.

[11] M. Cleves. *An Introduction to Survival Analysis Using Stata, Second Edition*. Stata Press, 2008. ISBN 1597180416. 372 pp. URL https://books.google.com/books?id=xttbn0a-QR8C&pgis=1.

[12] N. R. Cook. Use and misuse of the receiver operating characteristic curve in risk prediction. *Circulation*, 115(7):928–35, feb 2007. ISSN 1524-4539. URL http://www.ncbi.nlm.nih.gov/pubmed/17309939.

[13] T. Couture and Y. Gagnon. An analysis of feed-in tariff remuneration models: Implications for renewable energy investment. *Energy Policy*, 38(2):955–965, feb 2010. ISSN 03014215. URL http://linkinghub.elsevier.com/retrieve/pii/S0301421509007940.

[14] C. Davenport. The Marshall Islands Are Disappearing, dec 2015. URL http://www.nytimes.com/interactive/2015/12/02/world/The-Marshall-Islands-Are-Disappearing.html.

[15] F. W. Dekker, R. de Mutsert, P. C. van Dijk, C. Zoccali, and K. J. Jager. Survival analysis: time-dependent effects and time-varying risk factors. *Kidney International*, 74(8):994–997, 2008. ISSN 00852538.

[16] N. Dhere. Reliability of PV modules and balance-of-system components. In *Conference Record of the Thirty-first IEEE Photovoltaic Specialists Conference, 2005.*, pages 1570–1576. IEEE, 2005. ISBN 0-7803-8707-4. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1488445.

[17] L. E. Doman and E. al. International Energy Outlook 2016. Technical report, U.S. Energy Information Administration, Washington DC, 2016. 291 pp. URL http://www.eia.gov/forecasts/ieo/pdf/0484(2016).pdf.

[18] L. Duchateau and P. Janssen. *The frailty model*. Springer, New York, 2008. 329 pp.

[19] R. A. Feely, C. L. Sabine, J. M. Hernandez-Ayon, D. Ianson, and B. Hales. Evidence for upwelling of corrosive "acidified" water onto the continental shelf. *Science (New York, N.Y.)*, 320(5882):1490–2, jun 2008. ISSN 1095-9203. URL http://www.ncbi.nlm.nih.gov/pubmed/18497259.

[20] D. Feldman, G. Barbose, R. Margolis, T. James, S. Weaver, N. Darghouth, R. Fu, C. Davidson, S. Booth, and R. Wiser. Photovoltaic System Pricing Trends, 2014. URL http://www.nrel.gov/docs/fy14osti/62558.pdf.

[21] J. Flicker. PV Inverter Performance and Component-Level Reliability. Technical report, Sandia National Laboratories, Albuquerque, 2014. 23 pp. URL http://www.nrel.gov/pv/pdfs/2014_pvmrw_35_flicker.pdf.

[22] Frankfurt School. Global Trends in Renewable Energy Investment 2016. Technical report, United Nations Environment Programme, Frankfurt, 2016. URL http://fs-unep-centre.org/sites/default/files/publications/globaltrendsinrenewableenergyinvestment2016lowres_0.pdf.

[23] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*. CRC Press, Boca Raton, 3rd ed. edition, 2014. 656 pp. URL http://amstat.tandfonline.com/doi/full/10.1080/01621459.2014.963405.

[24] A. Gelman and D. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992. URL http://www.jstor.org/stable/2246093.

[25] J. Goldemberg, N. Unidas, P. d. l. N. U. para el Desarrollo., and W. E. Council. World energy assessment : energy and the challenge of sustainability. Technical report, United Nations Development Programme, New York, N.Y., 2000. ISBN 9211261260 9789211261264. 506 pp. URL http://www.undp.org/content/undp/en/home/librarypage/environment-energy/sustainable_energy/world_energy_assessmentenergyandthechallengeofsustainability.html.

[26] A. Golnas. PV System Reliability: An Operator's Perspective. *IEEE Journal of Photovoltaics*, 3(1):416–421, jan 2013. ISSN 2156-3381. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6305521.

[27] G. Grigoryan. Solar System Life Maintenance. Technical report, Pacific Gas & Electric Co., San Francisco, 2010. 16 pp. URL http://www.pge.com/includes/docs/pdfs/shared/solar/solareducation/solar_system_life_maintenance.pdf.

[28] T. Hals and N. Groom. Solar developer SunEdison in bankruptcy as aggressive growth plan unravels, apr 2016. URL http://www.reuters.com/article/us-sunedison-inc-bankruptcy-idUSKCN0XI1TC.

[29] F. Harrell. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer, New York, 2001. 582 pp.

[30] D. J. Hendry. Data generation for the Cox proportional hazards model with time-dependent covariates: a method for medical researchers. *Statistics in Medicine*, 33(3): 436–54, feb 2014. ISSN 1097-0258. URL http://www.ncbi.nlm.nih.gov/pubmed/24014094.

[31] Y. Hong, W. Q. W. Meeker, and J. J. D. Mccalley. Prediction of remaining life of power transformers based on left truncated and right censored lifetime data. *The Annals of Applied Statistics*, 3(2):857–879, 2009. URL http://www.jstor.org/stable/30244268.

[32] D. W. Hosmer, S. Lemeshow, and S. May. *Applied survival analysis: Regression modelling of time to event data*. John Wiley & Sons, Hoboken, second edi edition, 2008. 441 pp.

[33] J. Ibrahim, M. Chen, and D. Sinha. *Bayesian survival analysis*. Springer, New York, 2005. 493 pp. URL http://onlinelibrary.wiley.com/doi/10.1002/0470011815.b2a11006/full.

[34] IRENA. Battery Storage for Renewables: Market Status and Technology Outlook. Technical report, International Renewable Energy Agency, Masdar City, 2015.

60 pp. URL http://www.irena.org/documentdownloads/publications/irena_battery_storage_report_2015.pdf.

[35] IRENA. The Power to Change: Solar and Wind Cost Reduction Potential to 2025. Technical report, International Renewable Energy Agency, Masdar City, 2016. 112 pp. URL http://www.irena.org/DocumentDownloads/Publications/IRENA_Power_to_Change_2016.pdf.

[36] G. H. Jowett. The Exponential Distribution and Its Applications. *The Incorporated Statistician*, 8(2):89, jan 1958. ISSN 14669404. URL http://www.jstor.org/stable/10.2307/2986561?origin=crossref.

[37] L. Kang, W. Chen, N. A. Petrick, and B. D. Gallas. Comparing two correlated C indices with right-censored survival outcome: a one-shot nonparametric approach. *Statistics in medicine*, 34(4):685–703, feb 2015. ISSN 1097-0258. URL http://www.ncbi.nlm.nih.gov/pubmed/25399736 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4314453.

[38] J. P. Klein and M. L. Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. Springer, 2nd editio edition, 2003. 542 pp.

[39] D. G. Kleinbaum and M. Klein. *Survival Analysis: A Self-Learning Text*. Springer New York, second edi edition, 2005. ISBN 9780387239187.

[40] J. Kruschke. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Elsevier, London, 2nd ed. edition, 2015. 749 pp.

[41] H. Laukamp. Reliability study of grid connected PV systems: Field experience and recommended design practice. Technical report, Fraunhofer Institut fr Ware Energiesysteme, Freiburg, 2002. 39 pp. URL http://www.solelprogrammet.se/Global/Projekteringsverktyg/PDF/herman laukamp fin_rep.pdf.

[42] J. Levitt. *Complete guide to preventive and predictive maintenance*. Industrial Press, 2011. ISBN 9780831134419.

[43] D. Y. Lin and Z. Ying. Additive Hazards Regression Models for Survival Data. In D. Lin and T. Fleming, editors, *Proceedings of the First Seattle Symposium in Biostatistics: Survival Analysis*, pages 185–198. Springer US, Seattle, 1997. URL http://link.springer.com/10.1007/978-1-4684-6316-3_10.

[44] J. Marin and C. Robert. *Bayesian core: a practical approach to computational Bayesian statistics*. Springer, New York, 2007. 264 pp.

[45] A. Marshall and I. Olkin. *Life distributions*. Springer, New York, 2007. URL http://link.springer.com/content/pdf/10.1007/978-0-387-68477-2.pdf.

[46] C. Mason. Demography 213 Fall 2015, 2015. URL http://courses.demog.berkeley.edu/mason213F15/.

[47] G. Masson, S. Orlandi, and M. Rekinger. Global Market Outlook for Photovoltaics 2014-2018. Technical report, European Photovoltaic Industry Association, Brussels, 2014.

[48] R. McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC Press, Boca Raton, 2016.

[49] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087, 1953. ISSN 00219606. URL http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114.

[50] R. Mobley. *An introduction to predictive maintenance*. Elsevier Science, 2nd ed. edition, 2002. ISBN 0750675314. 447 pp.

[51] L. M. Moore and H. N. Post. Five years of operating experience at a large, utility-scale photovoltaic generating plant. *Progress in Photovoltaics: Research and Applications*, 16(3):249–259, may 2008. ISSN 10627995. URL http://doi.wiley.com/10.1002/pip.800.

[52] OECD. Solar Energy Perspectives. Technical report, International Energy Agency, Paris, dec 2011. ISBN 9789264124578. URL http://www.oecd-ilibrary.org/energy/solar-energy-perspectives_9789264124585-en.

[53] OECD. Technology Roadmap - Solar Photovoltaic Energy. Technical report, International Energy Agency, Paris, 2014. 60 pp. URL http://www.iea.org/publications/freepublications/publication/TechnologyRoadmapSolarPhotovoltaicEnergy_2014edition.pdf.

[54] OECD. World energy outlook 2015. Technical report, OECD, Paris, 2015. URL http://www.worldenergyoutlook.org/.

[55] M. Parry, C. Rosenzweig, and A. Iglesias. Effects of climate change on global food production under SRES emissions and socio-economic scenarios. *Global Environmental Change*, 14:53–67, 2004. URL http://www.sciencedirect.com/science/article/pii/S0959378003000827.

[56] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich. Smart Cooling of Data Centers. In *2003 International Electronic Packaging Technical Conference and Exhibition, Volume 2*, pages 129–137. ASME, jan 2003. ISBN 0-7918-3674-6. URL http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/IPACK2003-35059.

[57] G. Petrone, G. Spagnuolo, R. Teodorescu, M. Veerachary, and M. Vitelli. Reliability Issues in Photovoltaic Power Processing Systems. *IEEE Transactions on Industrial Electronics*, 55(7):2569–2580, jul 2008. ISSN 0278-0046. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4497162.

[58] J. B. Quinn and G. D. Quinn. A practical and systematic review of Weibull statistics for reporting strengths of dental materials. *Dental materials : official publication of the Academy of Dental Materials*, 26(2):135–47, feb 2010. ISSN 1879-0097. URL http://www.ncbi.nlm.nih.gov/pubmed/19945745 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3086645.

[59] H. Rinne. *The Weibull distribution: a handbook*. CRC Press, Boca Raton, 2008.

[60] G. Rodriguez. Lecture Notes on Generalized Linear Models, 2007. URL http://data.princeton.edu/wws509/notes/.

[61] S. Romdhane and L. Belkacem. Frailty Modeling for clustered survival data: a simulation study. In *International Pension Workshop*, page 22. International Actuarial Association, Paris, 2015. URL http://www.actuaries.org/oslo2015/papers/IAALS-Romdhane.pdf.

[62] Z. Schlanger. Fracking Wells Tainting Drinking Water in Texas and Pennsylvania, Study Finds, sep 2014. URL http://europe.newsweek.com/fracking-wells-tainting-drinking-water-texas-and-pennsylvania-study-finds-270735.

[63] SMA. Service and Maintenance. Technical report, SMA Solar Technology AG (DE), Niestetal, 2012. 76 pp.

[64] R. Solnit. Oil fuels war and terrorists like Isis., dec 2015. URL https://www.theguardian.com/commentisfree/2015/dec/08/oil-fuels-war-terrorists-isis-climate-movement-peace-cop-21.

[65] Stan Development Team. Documentation: Learn to use Stan, 2016. URL http://mc-stan.org/documentation/.

[66] Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual*. Stan, 2.10.0 edition, 2016. 576 pp. URL http://mc-stan.org.

[67] H. Steck, B. Krishnapuram, C. Dehing-oberije, P. Lambin, and V. C. Raykar. On Ranking in Survival Analysis: Bounds on the Concordance Index. In J. C. Platt and D. Koller and Y. Singer and S. T. Roweis, editor, *Advances in Neural Information Processing Systems 20*, pages 1209–1216. Curran Associates, Inc., Red Hook, 2008. URL http://papers.nips.cc/paper/3375-on-ranking-in-survival-analysis-bounds-on-the-concordance-index.pdf.

[68] M. Tableman and J. Sung. *Survival analysis using S: analysis of time-to-event data*. CRC Press, Boca Raton, 2004. ISBN 1584884088. 277 pp.

[69] D. K. Tatlow. China Air Quality Study Has Good News and Bad News, mar 2016. URL http://www.nytimes.com/2016/03/31/world/asia/china-air-pollution-beijing-shanghai-guangzhou.html.

[70] G. Tripepi, K. J. Jager, F. W. Dekker, and C. Zoccali. Statistical methods for the assessment of prognostic biomarkers (Part I): discrimination. *Nephrology, dialysis, transplantation : official publication of the European Dialysis and Transplant Association - European Renal Association*, 25(5):1399–401, may 2010. ISSN 1460-2385. URL http://www.ncbi.nlm.nih.gov/pubmed/20139066.

[71] M. Tso. Course Notes on Reliability 334, 2010. URL http://www.maths.manchester.ac.uk/~mkt/334 Reliability/Notes08/RelS2.pdf.

[72] F. B. Uz. Cortana Intelligence Solution Template Playbook for predictive maintenance in aerospace and other businesses, 2016. URL https://azure.microsoft.com/en-us/documentation/articles/cortana-analytics-playbook-predictive-maintenance/.

[73] A. Vehtari and A. Gelman. WAIC and cross-validation in Stan. *Submitted. http://www. stat. columbia.*, page 15, 2014. URL `http://dosen.narotama.ac.id/wp-content/uploads/2014/12/WAIC-and-cross-validation-in-Stan.pdf`.

[74] A. Vehtari, A. Gelman, and J. Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *arXiv:1507.04544v4*, jul 2015. URL `http://arxiv.org/abs/1507.04544`.

[75] S. Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3594, 2010. URL `http://www.jmlr.org/papers/v11/watanabe10a.html`.

[76] S. Weckx, C. Gonzalez, and J. Driesen. Reducing grid losses and voltage unbalance with PV inverters. In *2014 IEEE PES General Meeting — Conference & Exposition*, pages 1–5. IEEE, jul 2014. ISBN 978-1-4799-6415-4. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6939416`.

[77] A. Wienke. *Frailty models in survival analysis*. CRC Press, Boca Raton, 2010. ISBN 9781420073881.

[78] Wikipedia. List of oil spills, 2015. URL `https://en.wikipedia.org/wiki/List_of_oil_spills`.

[79] C. Wilson, A. Grubler, K. S. Gallagher, and G. F. Nemet. Marginalization of end-use technologies in energy innovation for climate protection. *Nature Climate Change*, 2(11):780–788, oct 2012. ISSN 1758-678X. URL `http://www.nature.com/doifinder/10.1038/nclimate1576`.

[80] H. Yu, H. Lo, H. Hsieh, and J. Lou. Feature engineering and classifier ensemble for KDD cup 2010. *Proceedings of the KDD Cup 2010 Workshop*, page 16, 2010. URL `http://mslab.csie.ntu.edu.tw/~mslab/publications/Conference/2011/Feature engineering and classifier ensemble for KDD Cup 2010.pdf`.

[81] Z. Žabokrtský. Feature Engineering in Machine Learning, 2015. URL `https://ufal.mff.cuni.cz/~zabokrtsky/courses/npfl104/html/feature_engineering.pdf`.

Leuven Statistics Research Centre (LStat)
Celestijnenlaan 200 B, bus 5307
3001 Heverlee, Belgium
tel. +32 16 32 88 75
www.kuleuven.be