Writeup

Yannik Pitcan
Panos Lambrianides
Ram Akella
Anil Aswani
Phil Kaminsky

July 22, 2019

Preface

In this document, we will discuss the types of datasets, distributions, and contexts we need to make inferences from them. The first purpose for this writeup is to demonstrate why distribution models and parameters are necessary. The second purpose is to layout a proposed research plan.

Introduction

We want to develop a probabilistic model for RUL (remaining useful life) of equipment, given historical run-to-failure data and past operational history.

As we view sensor data and not the machine states, one approach may be to understand the distribution of sensor data for a normal functioning machine and compare that to the distribution for one that's failing.

For our model, we have C-MAPSS inputs that simulate degradation scenarios, and then outputs to measure the system response. The fundamental flaw with past work is that the data generation process was ad-hoc and didn't assume any stochastic component for the flow and efficiency loss.

What To Improve

Currently, modeling of the RUL and FP is not distribution based. We think that there is much to be gained by understanding the distribution of failure times beyond those of point estimates. There are two ways we propose to do so:

- Parametric: Assume the distribution of failure times follows an inverse gamma distribution. Do a Bayesian update of this once observing our machine. This does assume knowledge about the distribution of observed values, but parameter updates for this are straight-forward.
- Nonparametric: Use a k-nearest-neighbors algorithm to model the failure times.

Fusion of Data from Two Distributions

This problem breaks down if we have two sets of observations.

Given a machine, we want to return a survival distribution dataset. The nuance here is we have two datasets, how do we combine these two datasets?

Without a prior distribution, one cannot blend the two observed datasets. Currently, the methods in place are RUL and FP point estimates.

We don't quite know how these numbers are generated. We cannot build models if we don't know what those numbers mean!

To delve into more detail, RUL is given by

$$\mathrm{E}[T-t|T>t] = \int_t^\infty zf(z|D)\,dz = \int_t^\infty zf(z)\,dz.$$

and FP is the cumulative probability up to time t given by

$$F(t) = \int_0^t f(\tau)\,d\tau.$$

How does one combine a point estimate value with observed data? This is not known, but it is doable to combine estimates when we have prior information about the distribution.

Furthermore, if I have both distributions with similar parametric form, then it's straightforward to update the prior with respect to each of the datasets and combine them. Note that the following examples are informal and not necessarily what we will use in our project. These are just used to demonstrate what we require beyond point estimates.

Simple Example with Beta-Binomial

For a simple example, let's first look at a case where we take draws from a binomial distribution $X \sim Bin(n,p)$ where $p \sim Beta(\alpha,\beta)$. Then one can update our knowledge about p.

$$f_{p|X}(p|X=x) \propto f_P(p)f_{X|p}(x|p) \propto p^{\alpha-1}(1-p)^{\beta-1}p^x(1-p)^{n-x} = p^{\alpha+x-1}(1-p)^{n-x+\beta-1}.$$

Here, we see that $p|X \sim Beta(\alpha+x, n-x+\beta)$. Similarly, if we have $Y \sim Bin(m,p)$, then $p|X,Y \sim Beta(\alpha+x+y, n-x-y+\beta)$. Then it's easy to get a new estimate for p. For instance, one can use a MAP estimation to get

$$\hat{p}_{MAP} = \frac{\alpha+x+y}{\alpha+\beta+n}.$$

Weibull-Inverse Gamma Demonstration

Now let's look at the case where we have Weibull and Inverse Gamma distributions.

The Weibull distribution is given by

$$f(x|\lambda,k) = \frac{k}{\lambda}x^{k-1}\exp\left\{-x^k/\lambda\right\}$$

Then

$$L(\lambda|x_1,\ldots,x_n) \propto \prod_i \lambda^{-1}\exp\left\{-\frac{x_i^k}{\lambda}\right\} = \lambda^{-n}\exp\left\{-\frac{\sum_i x_i^k}{\lambda}\right\}$$

The inverse gamma prior is

$$P(\lambda|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{-\alpha-1}\exp\left\{-\frac{\beta}{\lambda}\right\}$$

The posterior then is

$$P(\lambda|x_1,\ldots,x_n) \propto L(\lambda|x_1,\ldots,x_n)P(\lambda|\alpha,\beta) \propto \lambda^{-n}\exp\left\{-\sum_i x_i^k\lambda^{-1}\right\}\lambda^{-\alpha-1}\exp\left\{-\frac{\beta}{\lambda}\right\}$$
$$= \lambda^{-n-\alpha-1}\exp\left\{-\frac{\sum_i x_i^k+\beta}{\lambda}\right\}$$

One thus sees the posterior distribution of λ is inverse gamma with parameters $n+\alpha$ and $\sum_i x_i^k+\beta$. Subsequently, if we observe two datasets given by $x_1,\ldots,x_n$ and $y_1,\ldots,y_m$, then the update is simple and our posterior is still an inverse gamma with parameters $m+n+\alpha$ and $\sum_i x_i^k+\sum_j y_j^k+\beta$.

This update is relevant because the Weibull distribution is often used to model failure times of machines. Assuming the RUL and FP distributions are inverse gamma, then we have the above result. With the knowledge of how to update the distribution, it is more tractable to understand the RUL and FP updates fully.

Above, we used the assumption that the observations from different sources shared the same parametric model. But what if this were not the case? Then we have to do a bit of fitting. One could use a method of moments technique to fit a distribution to a data source that has the same parametric model and still carry the above steps. In particular, our constraints for method of moments techniques would be the point estimates provided. One may also utilize hierarchical Bayesian models here.

Estimating Parameters

Earlier, we hinted at how one can use the above updates to obtain new estimates of the underlying parameters. In this section, we discuss the MAP and MoM (Method of Moments) techniques in further detail.

MAP Estimates

Assume that we want to estimate an unobserved population parameter θ on the basis of observations x. Let f be the sampling distribution of x, so that $f(x \mid \theta)$ is the probability of x when the underlying population parameter is θ. Then the function:

$$\theta \mapsto f(x \mid \theta)$$

is known as the likelihood function and the estimate:

$$\hat{\theta}_{\mathrm{MLE}}(x) = \arg\max_{\theta} f(x \mid \theta)$$

is the maximum likelihood estimate of θ.

Now assume that a prior distribution g over θ exists. This allows us to treat θ as a random variable as in Bayesian statistics. We can calculate the posterior distribution of θ using Bayes' theorem:

$$\theta \mapsto f(\theta \mid x) = \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta}$$

where g is density function of θ, Θ is the domain of g.

The method of maximum a posteriori estimation then estimates θ as the mode of the posterior distribution of this random variable:

$$\hat{\theta}_{\mathrm{MAP}}(x) = \arg\max_{\theta} f(\theta \mid x) = \arg\max_{\theta} \frac{f(x \mid \theta)\, g(\theta)}{\displaystyle\int_{\Theta} f(x \mid \vartheta)\, g(\vartheta)\, d\vartheta} = \arg\max_{\theta} f(x \mid \theta)\, g(\theta).$$

The denominator of the posterior distribution (so-called marginal likelihood) is always positive and does not depend on θ and therefore plays no role in the optimization. Observe that the MAP estimate of θ coincides with the ML estimate when the prior g is uniform (that is, a constant function).

Method of Moments

This is a more frequentist in nature as it does not involve having prior parameter information.

Suppose that the problem is to estimate k unknown parameters $\theta_1, \theta_2, \ldots, \theta_k$ characterizing the probability distribution $f_W(w;\theta)$ of the random variable W. Suppose the first k moments of the true distribution (the "population moments") can be expressed as functions of the θs:

$$\mu_1 \equiv \mathrm{E}[W] = g_1(\theta_1, \theta_2, \ldots, \theta_k), \tag{0.1}$$
$$\mu_2 \equiv \mathrm{E}[W^2] = g_2(\theta_1, \theta_2, \ldots, \theta_k), \tag{0.2}$$
$$\vdots \tag{0.3}$$
$$\mu_k \equiv \mathrm{E}[W^k] = g_k(\theta_1, \theta_2, \ldots, \theta_k). \tag{0.4}$$

Suppose a sample of size n is drawn, resulting in the values $w_1, \ldots, w_n$. For $j = 1, \ldots, k$, let $\widehat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} w_i^j$ be the "j"-th sample moment, an estimate of μ_j. The method of moments estimator for $\theta_1, \theta_2, \ldots, \theta_k$ denoted by $\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_k$ is defined as the solution (if there is one) to the equations

$$\widehat{\mu}_1 = g_1(\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_k), \tag{0.5}$$
$$\widehat{\mu}_2 = g_2(\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_k), \tag{0.6}$$
$$\vdots \tag{0.7}$$
$$\widehat{\mu}_k = g_k(\widehat{\theta}_1, \widehat{\theta}_2, \ldots, \widehat{\theta}_k). \tag{0.8}$$

Hierarchical Bayesian Models

Hierarchical bayesian models are multilevel statistical models that allow us to incorporate richer information into the model. Note that this concept ties in closely to the MAP parameter estimation as it gives us a way of constructing the posterior distribution to optimize.

Components

Bayesian hierarchical modeling makes use of two important concepts in deriving the posterior distribution, namely:

- Hyperparameters: parameters of the prior distribution
- Hyperpriors: distributions of Hyperparameters

Suppose a random variable Y follows a normal distribution with parameter θ as the mean and 1 as the variance, that is $Y \mid \theta \sim N(\theta, 1)$. Suppose also that the parameter θ has a distribution given by a normal distribution with mean μ and variance 1, i.e. $\theta \mid \mu \sim N(\mu, 1)$. Furthermore, μ follows another distribution given, for example, by the standard normal distribution, $\mathrm{N}(0, 1)$. The parameter μ is called the hyperparameter, while its distribution given by $\mathrm{N}(0, 1)$ is an example of a hyperprior distribution. The notation of the distribution of Y changes as another parameter is added, i.e. $Y \mid \theta, \mu \sim N(\theta, 1)$. If there is another stage, say, μ follows another normal distribution with mean β and variance ϵ, meaning $\mu \sim N(\beta, \epsilon)$, β and ϵ can also be called hyperparameters while their distributions are hyperprior distributions as well.

Framework

Let y_j be an observation and θ_j a parameter governing the data generating process for y_j. Assume further that the parameters $\theta_1, \theta_2, \ldots, \theta_j$ are generated exchangeably from a common population, with distribution governed by a hyperparameter ϕ.

The Bayesian hierarchical model contains the following stages:

$$\text{Stage I: } y_j \mid \theta_j, \phi \sim P(y_j \mid \theta_j, \phi)$$

$$\text{Stage II: } \theta_j \mid \phi \sim P(\theta_j \mid \phi)$$

$$\text{Stage III: } \phi \sim P(\phi)$$

The likelihood, as seen in stage I is $P(y_j \mid \theta_j, \phi)$, with $P(\theta_j, \phi)$ as its prior distribution. Note that the likelihood depends on ϕ only through θ_j.

The prior distribution from stage I can be broken down into:

$$P(\theta_j, \phi) = P(\theta_j \mid \phi)P(\phi)$$

from the definition of conditional probability, with ϕ as its hyperparameter with hyperprior distribution, $P(\phi)$.

Thus, the posterior distribution is proportional to:

$$P(\phi, \theta_j \mid y) \propto P(y_j \mid \theta_j, \phi)P(\theta_j, \phi)$$

using Bayes' Theorem.

$$P(\phi, \theta_j \mid y) \propto P(y_j \mid \theta_j)P(\theta_j \mid \phi)P(\phi)$$

Future Steps

There are several open problems to be addressed here:

a) Solving the preventative maintenance optimization problem for a given distribution.

b) How can we combine two distributions? If we have multiple distributions for failure times, how can we choose a combination of these to combine with preventative maintenance optimization?

Current Direction

The problem can be split into two parts:
The first part is a preliminary problem that may be a paper of its own, but not the utmost task.

- Estimation of underlying states
- Using a POMDP solver to determine optimal actions.

For the first step (estimation), we used a variant of EM (expectation maximization) algorithm to determine underlying parameters. A DLM (dynamic linear model) was used for this purpose since we need to estimate the underlying states. This is necessary for formulating a POMDP that we can solve.

DLM model

The DLM used assumes our states correspond to different levels of system health. Originally we just used two states corresponding to 0 (failing) or 1 (healthy), but this model failed to capture some underlying information. Instead, we work with four states, where the two middle states represent the system in below average and above average condition, and the other two again representing failing and healthy condition.

Mathematically, this is represented as follows:

$$Ax_t + \epsilon_t = y_t$$

$$Bx_t + \psi_t = x_{t+1}$$

where ϵ_t and ψ_t are Gaussian draws from $N(0, \Sigma_O)$ and $N(0, \Sigma_S)$, where Σ_O and Σ_S are the covariance matrices for observations and state transitions respectively.

1 POMDP formulation

Switching gears, we can instead think of this as an HMM with noisy observations. The key problems we are tackling are the following:

- What is the optimal policy that reduces costs of repair?
- How can we assess which process generates our observations (i.e. is there a control policy in place or not)?

The first question, we can understand by treating this as a POMDP where we have state uncertainty.

1.1 State and observation definitions

Define our four states as $1, 2, 3, 4$ where 1 denotes poor condition and 4 represents excellent condition. State values 2 and 3 represent below average and above average respectively.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ .8 & .2 & 0 & 0 \\ .6 & .2 & .2 & 0 \\ .3 & .3 & .3 & .1 \end{bmatrix}$$

is our state transition matrix, where $p_{11} = 1$ and $p_{4,4} = .1$. These are arbitrary values in our transition matrix, but the point is that this is lower-triangular and our condition can never increase without intervention (control policy). Also we expect the most likely transition to go from s to $s-1$ for a state s. If we're already in the worst state 1, we stay there.

The observed measurements at state s (o_s) follows a Gaussian distribution with $o_s \sim N(\beta s, \Sigma_s)$, where $\beta \in \mathbb{R}^4$ is a fixed but unknown constant and $\Sigma_s^{n \times n}$ is a fixed unknown covariance matrix. We can learn both of these via EM methods if we generate enough observation measurements.

1.2 Reward Function

For our preventative maintenance problem, the cost function

$$R(s,a) = \begin{cases} e^s, a = 1 \\ e^{-s}, a = 0 \end{cases} = e^{(2a-1)s}$$

This can be changed, but the fundamental point is that we incur more cost when repairing in a state of high health or if we don't repair an unhealthy machine.

1.3 Control Policy

The control policy U we use is a threshold that repairs our system if we believe reach a state with low health (state 1 and 2). Mathematically, this is described by the following:

$$U(s) = \begin{cases} 1, & s = 1, 2 \\ 0, & otherwise \end{cases}$$

where 1 denotes repair, 0 denotes no action.

The transition probability matrix under a control policy presuming the transition matrix for no control is like before is

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & .2 & .8 \\ 0 & 0 & .3 & .1 \end{bmatrix}.$$

1.4 Mathematical Formulation of Optimal Policy

Define $U(o, a)$ to be the expected utility function corresponding to an action taken under a given observation. If ϕ denotes the mapping between observations to actions, the optimal ϕ is represented by

$$\phi(o) = \max_a \left[E[U(o, a)|s = 1] * P(s = 1|o) + \cdots + E[U(o, a)|s = 4] * P(s = 4|o) \right]$$

In our case, this becomes an maximization problem w.r.t a of

$$R(s = 1, a) * P(s = 1|o) + \cdots + R(s = 4, a) * P(s = 4|o) \Rightarrow$$

$$e^{(2a-1)} * P(s = 1|o) + \cdots + e^{(2a-1)*4} * P(s = 4|o).$$

We can write out $P(s = 1|o)$ explicity for the Gaussian observation case, but for our purposes, we'll keep this more general.

1.5 Belief state updates

The reason we use a POMDP framework as opposed to an MDP framework is because we have uncertainty regarding what state we are in at each timestep. Thus we have "belief states" that prescribe probability distributions for our states.

Let's write out how the policy updates belief-states for a POMDP.

Every time we execute an action $a = \pi(b)$ and observe o and reward r, we update our belief state b by $b := UpdateBelief(b, a, o)$. The following explains how the state transition function is calculated.

$$\begin{aligned} \tau(b'|b, a) &= P(b'|b, a) \\ &= \sum_o P(b'|b, a, o) P(o|b_t, a_t) \\ &= \sum_o P(b'|b, a, o) \sum_{s'} P(o|b, a, s') P(s'|b, a) \\ &= \sum_o P(b'|b, a, o) \sum_{s'} O(o|s') \sum_s P(s'|b, a, s) P(s|b, a) \\ &= \sum_o P(b'|b, a, o) \sum_{s'} O(o|s') \sum_s T(s'|s, a) b(s) \end{aligned}$$

$P(b'|b, a, o) = \delta_{b'}(UpdateBelief(b, a, o))$.

For belief updating in a discrete state space, one can use recursive Bayesian estimation. The new belief state b' is given by

$$\begin{aligned} b'(s') &= P(s'|o,a,b) \\ &\propto P(o|s',a,b)P(s'|a,b) \\ &\propto O(o|s',a)P(s'|a,b) \\ &\propto O(o|s',a)\sum_s P(s'|a,b,s)P(s|a,b) \\ &\propto O(o|s',a)\sum_s T(s'|s,a)b(s). \end{aligned}$$

Since we are working with four states, we can use the above for belief updates.

The next task is determining the optimal policy for this POMDP. One way of solving this in the general setting is via point-based value iteration.

1.6 Solving Using Point-Based Value Iteration

An alpha vector α is such that $U(b) = \alpha^T b$, where U is the expected utility function.

Let $B = b_1, \ldots, b_n$ denote the set of belief points and $\Gamma = \alpha_1, \ldots, \alpha_n$ the associated alpha vectors.

The value function at a new point b is estimated as follows:

$$U^\Gamma(b) = \max_{\alpha\in\Gamma} \alpha^T b = \max_{\alpha\in\Gamma} \Sigma_s \alpha(s)b(s).$$

Initialize all the alpha vectors to

$$\max_a \sum_{t=0}^{\infty} \gamma^t \min_s R(s,a) = \frac{1}{1-\gamma} \max_a \min_s R(s,a).$$

Update the value function at belief b based on the n alpha vectors

$$U(b) := \max_a \left[R(b,a) + \gamma \sum_o P(o|a,b)U(b') \right],$$

where b' is determined by UPDATEBELIEF(b,a,o) and

$$P(o|b,a) = \sum_s b(s) \sum_{s'} O(o|s',a)T(s'|s,a).$$

From Bayes' rule,

$$b'(s') = \frac{O(o|s',a)}{P(o|b,a)} \sum_s T(s'|s,a)b(s).$$

Last, combining these, we get

$$U(b) := \max_a \left[R(b,a) + \gamma \sum_o \max_{\alpha\in\Gamma} \sum_s b(s) \sum_{s'} O(o|s',a)T(s'|s,a)\alpha(s') \right].$$

Note that this is for the case where we have estimates of our underlying transition probabilities in our model.

1.7 Which Process Generates Our Data?

The second problem to think about is how one can determine whether our observations indicate the use of a control policy or not. One method involves the use of a sequential likelihood ratio test where we have the following:

Let $o_1, \ldots, o_n$ be a stream of measurement vectors for our system.

If we are trying to assess whether these measurements come from a system with a control policy or not, we can use a sequential test.

There is a caveat here, which is that our observation distribution parameters are unknown. We could use the parameter estimates but they change over time. Thus at timesteps u and t, $l_1(X_1^{(t)}) \neq l_1(X_1^u)$ and similarly for $l_0(X_0)$. One workaround for this is via collecting enough data until we can use an EM method to estimate β_s, σ_s for each state s. Then we could use the sequential probability ratio test with these parameters. This may be very computationally intensive, however. Another issue with the above is that an uncontrolled system will eventually reach an absorption state (system failure) and then it's intractable.

The alternative route would be to avoid parameter estimation altogether and work with objective functions to assess which distribution our data is coming from. The objective function encapsulates enough information about our underlying distributions for it to act as a means of identifying our data generation process. Reinforcement learning can be of use because we can avoid dealing with too complex data by only considering data points corresponding to high value functions or high probabilities.

READ THIS SECTION

To summarize, the two problems we want to solve are

- Can we use RL approaches to determine information about the time to failure (RUL) without resorting to parametric estimation?
- How can we understand optimal repair methods when we blend prior knowledge of our machine data? For example, do observed system measurements indicate a control policy is in place?

Experiment 1

As before, we create a simulation for a machine running until it reaches failure. The underlying states $(1, 2, 3, 4)$ represent the health level of the machine, with 1 indicating very-low/failing and 4 very-high/perfect condition. A level of 2 denotes low but not failing and 3 moderately high health.

Our state transitions are given by

$$\begin{bmatrix} p_{11} & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 \\ 0 & p_{32} & p_{33} & 0 \\ 0 & 0 & p_{43} & p_{44} \end{bmatrix}$$

where p_{ij} denotes the probability of the system health moving from level i to j. The above is for the case where no action is taken.

The actions here are either to repair $a = 1$ or do nothing $a = 0$.

When a repair is done, then

$$\begin{bmatrix} 0 & p'_{12} & p'_{13} & p'_{14} \\ 0 & 0 & p'_{23} & p'_{24} \\ 0 & 0 & 0 & p'_{34} \\ 0 & 0 & 0 & p'_{44} \end{bmatrix}$$

is our transition matrix, where we use p' to denote these are probabilities when we do a repair.

Our observations depend on whether we repair the system or not and for simplicity, assume the observed measurement follows a Gaussian distribution centered at the underlying health state with variance σ^2.

If we don't repair the system, then define $O(o|s, a = 0) \sim N(s, \sigma^2)$. Else, if we do repair, then $O(o|s, a = 1) \sim N(4, \sigma^2)$.

If we use discrete observation buckets, then

$$O(0) = \begin{bmatrix} NoRepair & 1 & 2 & 3 & 4 \\ 1 & o_{11} & o_{12} & o_{13} & o_{14} \\ 2 & o_{21} & o_{22} & o_{23} & o_{24} \\ 3 & o_{31} & o_{32} & o_{33} & o_{34} \\ 4 & o_{41} & o_{42} & o_{43} & o_{44} \end{bmatrix}$$

and

$$O(1) = \begin{bmatrix} Repair & 1 & 2 & 3 & 4 \\ 1 & o'_{11} & o'_{12} & o'_{13} & o'_{14} \\ 2 & o'_{21} & o'_{22} & o'_{23} & o'_{24} \\ 3 & o'_{31} & o'_{32} & o'_{33} & o'_{34} \\ 4 & o'_{41} & o'_{42} & o'_{43} & o'_{44} \end{bmatrix}$$

Lastly, our reward function takes the following form:

$$c(4 - s) + \sum_{s=1}^{4} I_M(a = 1|s)$$

where I is an indicator function. c is a negative constant so this represents increasing (negative) rewards with respect to the state, if there is no action taken and vice versa (decreasing, positive w.r.t the state)if there is an action.

Another way of writing this is as follows

$$R(0) = \begin{bmatrix} NoRepair & Reward \\ 1 & r_1 \\ 2 & r_2 \\ 3 & r_3 \\ 4 & r_4 \end{bmatrix}$$

$$R(1) = \begin{bmatrix} Repair & Reward \\ 1 & r'_1 \\ 2 & r'_2 \\ 3 & r'_3 \\ 4 & r'_4 \end{bmatrix}$$

We can define this reward function as $R(a) = c(4 - s) + \sum_{s=1}^{4} I(a = 1|s)$ where $R_s(0) = r_s$ and $R_s(1) = r'_s$.

The overarching goal is to use reinforcement learning methods with POMDPs to give an optimal strategy for repairing the system that minimizes the repair costs. And we believe that, in solving the POMDP, we can determine the time of failure implicity with a good choice of an objective function.

The two routes are as follows:

- Solving this machine failure model as a POMDP using value-iteration
  - In the discrete observation setting, we bucket our CMAPPS measurement observations into four groups.
- Model-free approaches that circumvent guesses of the underlying state transition matrices.

Experiment 2

Here, we would like to learn optimal policies for repairing a system when we have assumptions on the data generating process.

- No prior distribution – our data is generated via bootstrapping.
- One prior
  - Do we trust this prior distribution?
- We have two prior distributions and we would like to understand which one generates our data, or how they interact with each other. For example, we can combine two priors via a "weighted average" (partial update) or fully via a Bayesian update.

Appendix: End Of Life Analysis Background

The goal of the preliminary analysis is to create a Bayesian probabilistic survival model, using incomplete data sets in a general way. The goal of this section is to define the model and methodology used and to outline options and identify risk areas.

Let T be the continuous nonnegative random variable representation the end of life time of a machine in some population. Let f(t) be the pdf of T and let the distribution function be

$$F(t) = P(t \leq t) = \int_0^t f(u)du \tag{1.1}$$

The probability of a machine functioning until time t is given by the function

$$S(t) = 1 - F(t) = P(T > t) \tag{1.2}$$

where S(0)=1 and $S(\infty) = \lim_{t\to\infty} S(t) = 0$. The hazard function h(t) is defined as the instantaneous rate of failure at time t

$$h(t) = \frac{f(t)}{S(t)} \tag{1.3}$$

The interpretation is that $h(t)\Delta t$ is the approximate probability of failure in $(t, t + \Delta t)$. It is easy to show that

$$h(t) = -\frac{d}{dt} log(S(t)) \tag{1.4}$$

Integrating both sides and exponentiating we get

$$S(t) = exp(-\int_0^t h(u)du) \tag{1.5}$$

The hazard function H(t) and the survivor function S(t) are related by

$$S(t) = exp(-H(t)) \tag{1.6}$$

and the hazard function h(t) has the properties

$$h(t) \geq 0 \quad \text{and} \quad \int_0^\infty h(t)dt = \infty \tag{1.7}$$

Finally the survival pdf is given by

$$f(t) = h(t)exp(-\int_0^t h(u)du) \tag{1.8}$$

The Weibull distribution

The Weibull distribution is one of the most commonly used distributions to model survival pdf, and is the distribution we will use to model survival pdf

$$f(t) = \begin{cases} \alpha\gamma t^{\alpha-1} exp(-\gamma t^{\alpha}) & t > 0, \alpha > 0, \gamma > 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{1.9}$$

The Weibull distribution is denoted by $\mathcal{W}(\alpha, \gamma)$ where α and γ are the parameters of the distribution. For this distribution the hazard distribution h(t) is monotonically increasing when $\alpha > 1$ and monotonically decreasing when $0\alpha < 1$. It follows that the survivor function is

$$S(t) = exp(-\gamma t^{\alpha}) \tag{1.10}$$

and the hazard function is

$$h(t) = \gamma\alpha t^{\alpha-1} \tag{1.11}$$

and the cumulative hazard function H(t) is given by

$$H(t) = \gamma t^{\alpha} \tag{1.12}$$

Proportional Hazard Model

The hazard function depends on both time and a set of covariates. The proportional hazards model separates these components by by specifying that the hazard at time t, for a machine whose covariate vector is $\mathbf{x}$ is given by the linear relatioship

$$h(t|\mathbf{x}) = h_0(t) exp(\mathbf{x}'\beta) \tag{1.13}$$

where β is a vector of regression coefficients.

Censoring

We expect the end of life sensor data is right censored, that is, the survival times are only known for a portion of machines under study. The likelihood function for righ censored data will be constructed as follows. Suppose there are n machines and associated with the i^{th} individual is a survival time t_i and a fixed censoring time c_i. The t_o are assumed to be independent and identically distributed with density f(t) and survival function S(t). The exact survival time for machine i, t_i, will be observed only if $t_i \leq c_i$, The data can be represented as the n pairs of random variables (y_i, ν_i) where

$$y_i = \min(t_i, c_i) \tag{1.14}$$

and

$$\nu_i = \begin{cases} 1 \quad \text{if} \quad t_i \leq c_i, \\ 0 \quad \text{if} \quad t_i > c_i \end{cases} \tag{1.15}$$

Then the likelihood function for $(\beta, h_0(.))$ for a set of right censored data is given by

$$\begin{aligned} L(\beta, h_0(t)|D) \propto \prod_{i=1}^{n} [h_o(y_i) exp(\eta_i)]^{\nu_i} (S_0(y_i)^{exp(\eta_i)}) \\ \prod_{i=1}^{n} [h_o(y_i) exp(\eta_i)]^{\nu_i} exp\left\{ -\sum_{i=1}^{n} exp(\eta_i) H_0(y_i) \right\} \end{aligned} \tag{1.16}$$

where $D = (n, \mathbf{y}, X, \nu)$, $\mathbf{y} = (y_1, y_2, ..., y_n)'$ and $\nu = (\nu_1, \nu_2, ..., \nu_n)$, $\eta_i = \mathbf{x}_i'\beta$ is the linear predictor of machine i, and

$$S_0(t) = exp(-\int_0^t h_0(u)du) = exp(-H_0(t)) \tag{1.17}$$

is the baseline survivor function.