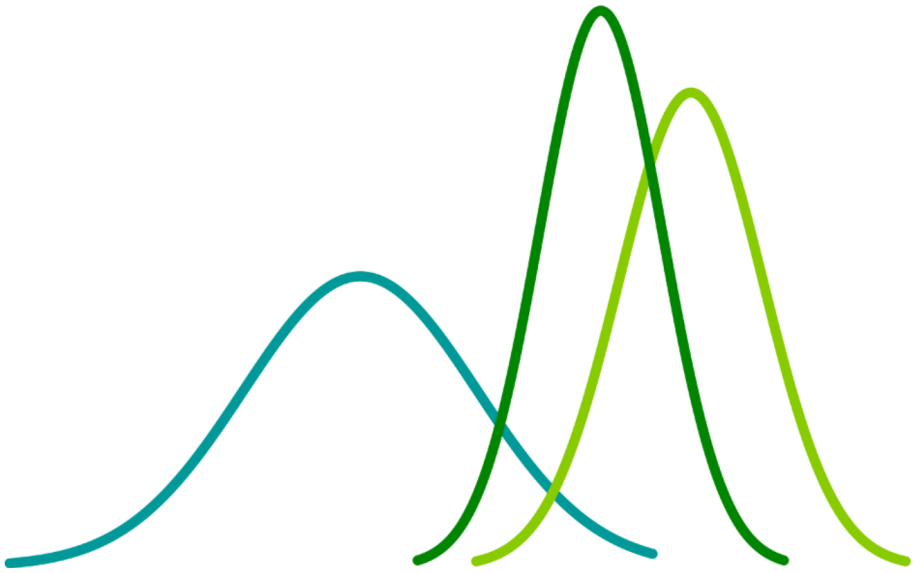
Bayesian Machine Learning

Bayesian Modelling of Component Degradation

University of Bath



Number Pages: 9
Word Count: 1488

Table of Contents

List of Figures

Consultancy Report for Client

1.1 Introduction

In modern mechanical systems, predicting component degradation is important to maintaining reliability and minimizing downtime for a company. In this project, we will try to predict how the integrity of mechanical components, termed "K", degrade over time. Our client manufactures and operates machinery that depends on these components. However, due to wear and tear during operation, the integrity of each K component degrades over time. To monitor this degradation, the client records an "integrity" measurement for each component, measured on a scale from 0 to 100, where 100 would represent full functionality and 0 represents component failure.

We have three client objectives for this project:

- To accurately model the degradation of component integrity over time using Bayesian techniques.
- To predict future integrity, especially for components with sparse data on them.
- To find the possible causes of degradation by analysing the features given by the client.

Traditional machine learning methods are known to perform poorly in cases like this where there is not much data available. Thus, in order to make the most of the available data and accurately quantify uncertainty, hierarchical Bayesian modelling was chosen, using Hamiltonian Monte Carlo techniques with NumPyro.

1.2 Baseline Model

1.2.1 Baseline Model Structure

The baseline model assumes that each component's integrity $y_i(t)$ over time t follows the function:

$$y_i(t) = f_i(t) + \epsilon, with,$$
$$f_i(t) = u_i \exp(-v_i t/100)$$

- Where u_i is the initial integrity intercept for component i,
- v_i is the decay rate for component i,
- ϵ is additive Gaussian noise with unknown standard deviation σ_y.

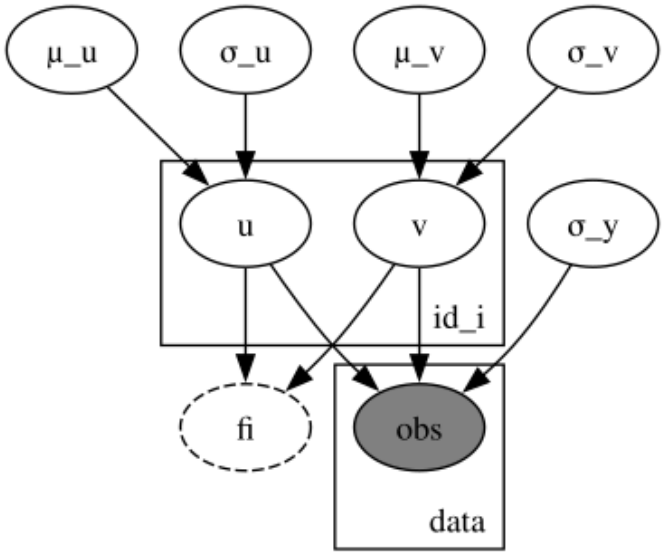


Figure 1.1: Baseline Model Rendered.

In the implementation, the model first defines hyper priors for the intercepts (μ_u, σ_u) and decay rates (μ_v, σ_v), using Normal and Half Normal distributions. The component specific parameters u_i and v_i are drawn from these hyper priors using `numpyro.plate`. The observation noise standard deviation σ_y also takes a Half Normal prior. The function $f_i(t)$ is calculated using the formula given by the client above. Finally, the observed data points are modelled as samples from a Normal distribution centred at $f_i(t)$ with variance σ_y^2. HMC inference was then used with the NUTS Sampler with 4 chains, 2000 warm-up steps, and 2000 samples per chain.

1.2.2 Baseline Model Posterior Diagnostics

The trace plots of the baseline model (shown in Figure 1.2) show healthy sampling behaviour. Each parameter shows smooth, stationary traces with good mixing across all of the chains. The posterior densities for these parameters are approximately Gaussian, centred around the sensible values. You can see that μ_u centres around 90, which makes sense as initial integrities are high. Similarly, μ_v centres around 4, showing moderate decay rates. u_i and v_i also show the posterior distributions do not really overlap, meaning the model can actually distinguish between all of the components. The effective sample sizes are high, $\hat{r}$ values are 1.00 for all parameters and there were no divergences.

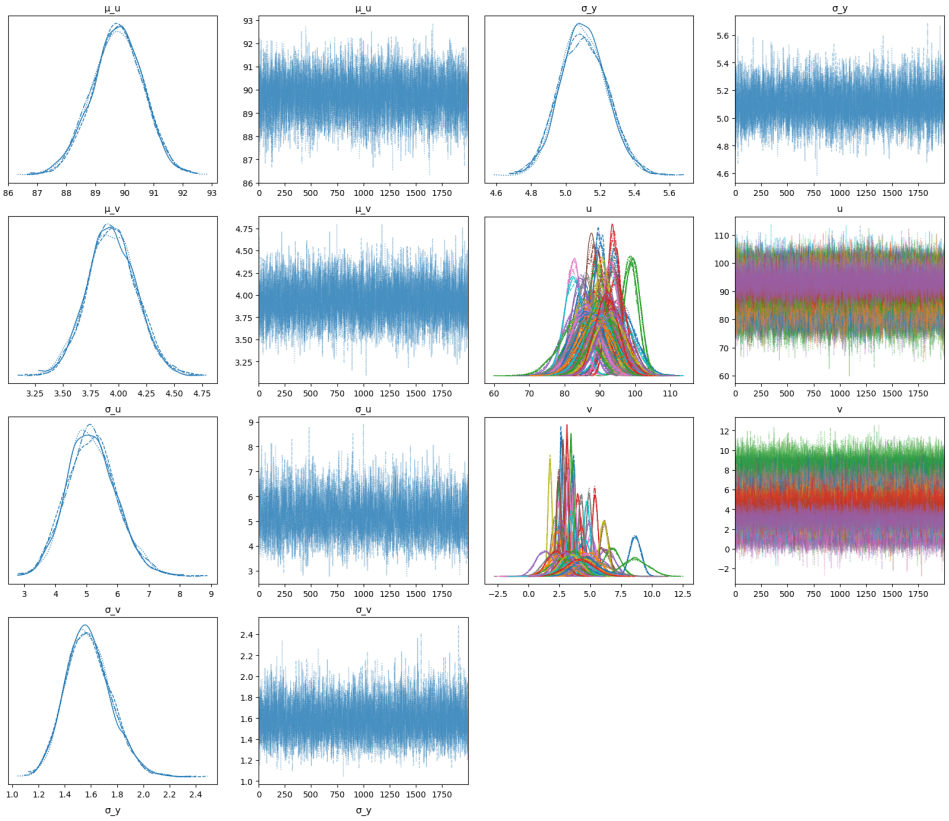


Figure 1.2: Baseline Model Trace Plots.

1.2.3 Baseline Model Predictive Performance

To evaluate the baseline model, posterior predictive plots were generated for four components, one with dense data, and three with sparse data (Figure 1.3). Each plot includes the observed data points, the posterior mean, and 2 error bars, one for observation with noise and one for without the noise. For components with lots of data available, the model successfully captures the downward trend of that components integrity, with its error bars tight around the posterior mean line. But for the components with spare data available, the predictions become more uncertain. The error bars for these components become really wide, showing that the model cannot infer decay without sufficient data. This makes sense as the model has less information to estimate u_i and v_i. In these cases, the model relies more on the hyper priors and the behaviour of other components, resulting in more variance.

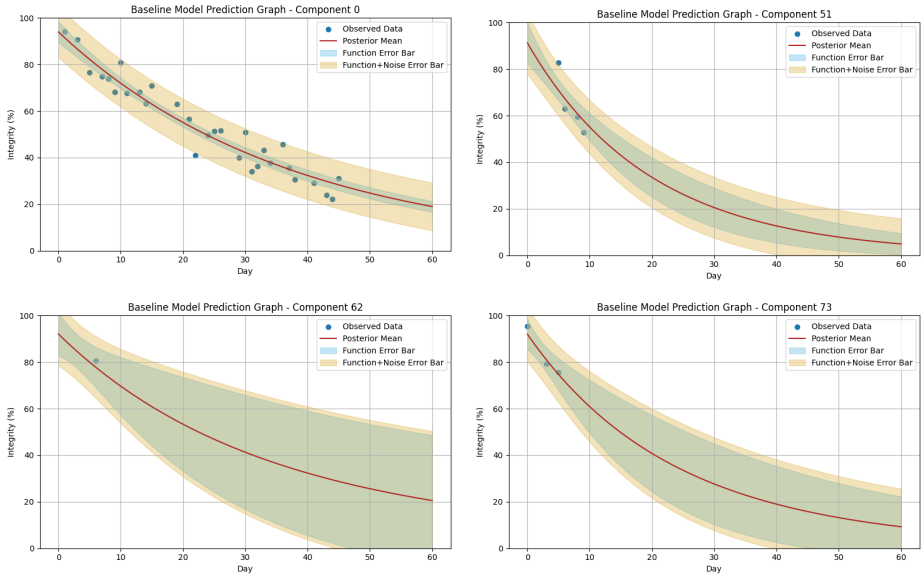


Figure 1.3: Baseline Model Prediction Graphs.

1.3 Enhanced Model

1.3.1 Enhanced Model Structure

The enhanced model modifies the degradation rate by incorporating linear effects of the diagnostic features. It uses the forumla:

$$f_i(t) = u_i \exp\left(-(v_i + \sum_{j=1}^{5} w_j x_{ji})t/100\right)$$

- Where w_j are weights shared across all components,
- x_{ji} are the diagnostic values for component i.

Including these diagnostics could explain the variability in degradation patterns, which is one of our client objectives.

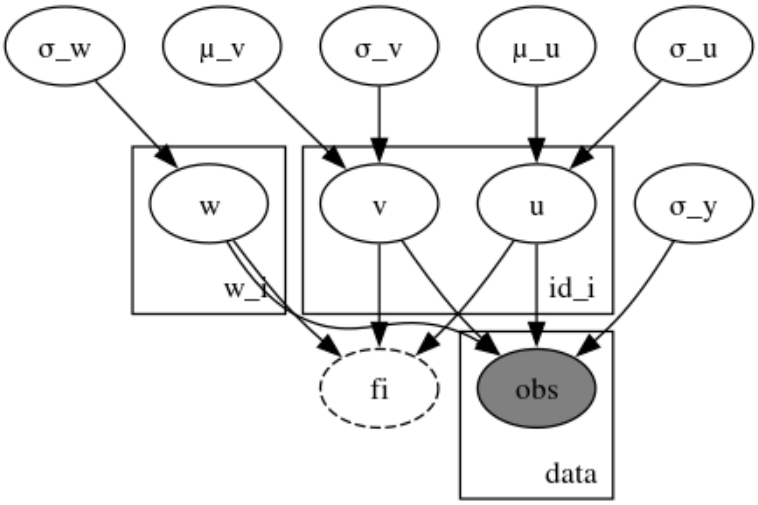


Figure 1.4: Enhanced Model Rendered.

In the code implementation, the enhanced model adds a new parameter for the weights w_j for $j = 1, \ldots, 5$. These weights model the effect of $X1$–$X5$ on the degradation rate. A Half Normal hyperprior is used for the weight σ_w, and each w_j is sampled from a zero mean Normal distribution with scale σ_w. Since v_i is adjusted by $\sum_{j=1}^{5} w_j x_{ji}$, the degradation rate is no longer dependent only on v_i, but also is influenced by the component diagnostics. The rest of the model structure and HMC sampling method remains unchanged from the baseline model.

1.3.2 Enhanced Model Posterior Diagnosis

Something interesting to note is that in the enhanced model, most diagnostic features showed weights centred near zero, meaning they have little effect on component degradation. But, as seen in Figure 1.5, the green curve has its weight centred near 1.0. This indicates that this feature is a strong predictor of the degradation rate, and suggests that differences in this measurement will affect component reliability significantly. Thus to increase the component lifespan, the client should look into managing and improving this feature.

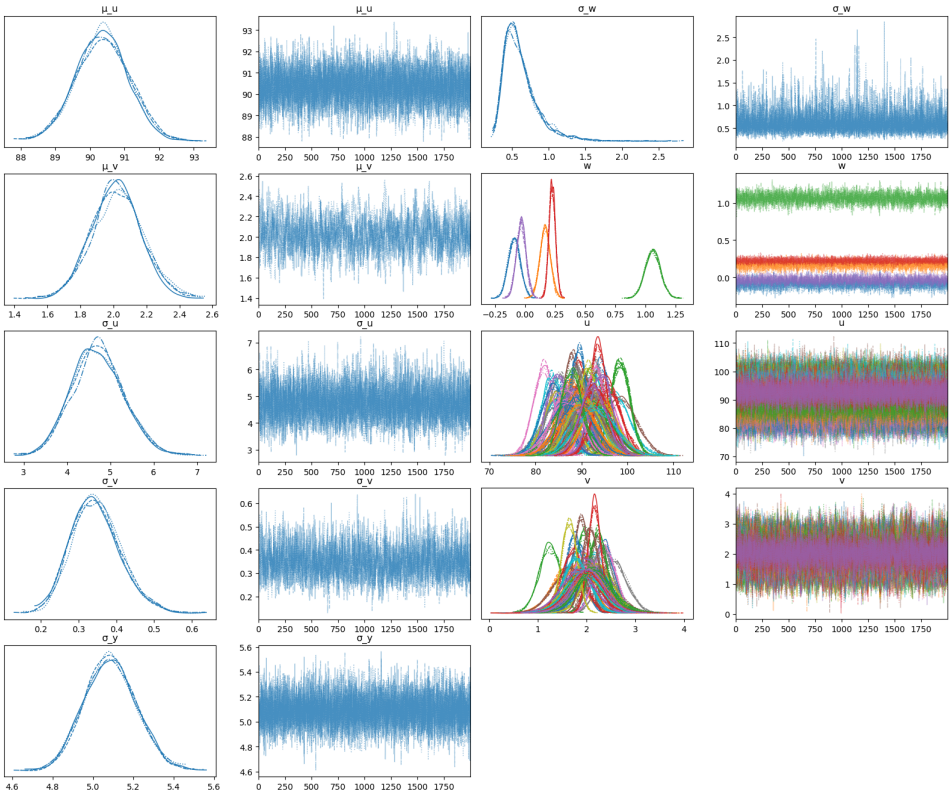


Figure 1.5: Baseline Model Trace Plots.

1.3.3 Enhanced Model Predictive Performance

Since the enhanced model uses additional component-specific characteristics, it is more precise in its predictions, especially for the components with less data available to it. As shown in Figure 1.6, the posterior mean curves are close to the observed data, and the error bands are noticeably narrower compared to the baseline model. This shows that the enhanced model shares information across the 5 components through the learned feature weightings well. This is why that compared to the baseline model, the enhanced model shows reduced uncertainty even when there is not much data available.

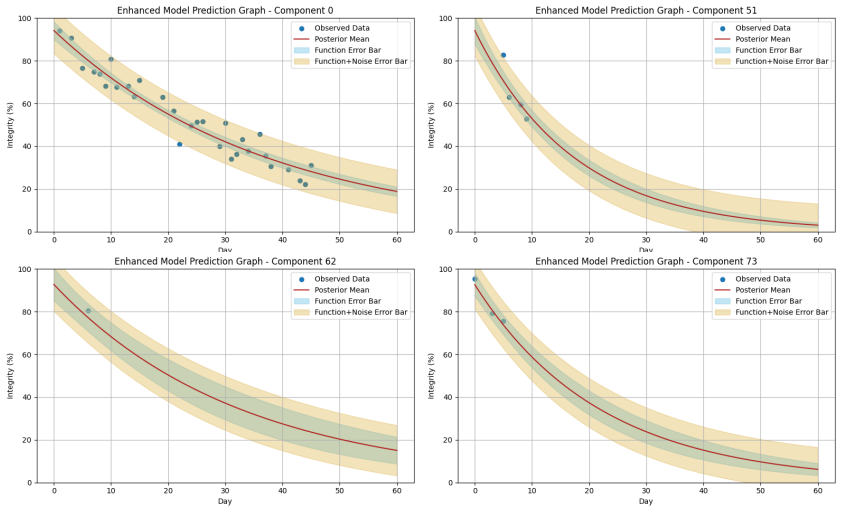


Figure 1.6: Enhanced Model Prediction Graphs.

1.3.4 Blind Test Results

Using the enhanced model, we generated predictions for components with less data (indices 50–7 4). For each component the posterior predictive mean and standard deviation at day 30 were calculated. The probability that integrity less than 30% was calculated using the CDF of a Gaussian with the predicted mean and standard deviation. The results were tabulated into a csv file and will help the client to validate the model using their actual data once it becomes available.

	ID	index	probability
0	K#0050	50	0.267128
1	K#0051	51	0.993183
2	K#0052	52	1.000000

Figure 1.7: Blind test results for three components.

1.4 Engineering Team Queries

1.4.1 Black Box Model

As requested by the client, we developed a black box model to compare results to our Bayesian models. For this task we implemented a linear regression model using sklearn, trained on features $X1$–$X5$, and some extra features we engineered. These features were the average integrity over days 0-10, log avergage integrity, the standard deviation of integrity, and the average gradient of integrity. This model was used to predict the integrity at day 30. We then plotted a graph showing how the black box model performed on its test split, comparing its results to the enhanced model on the same test components (Figure 1.8).

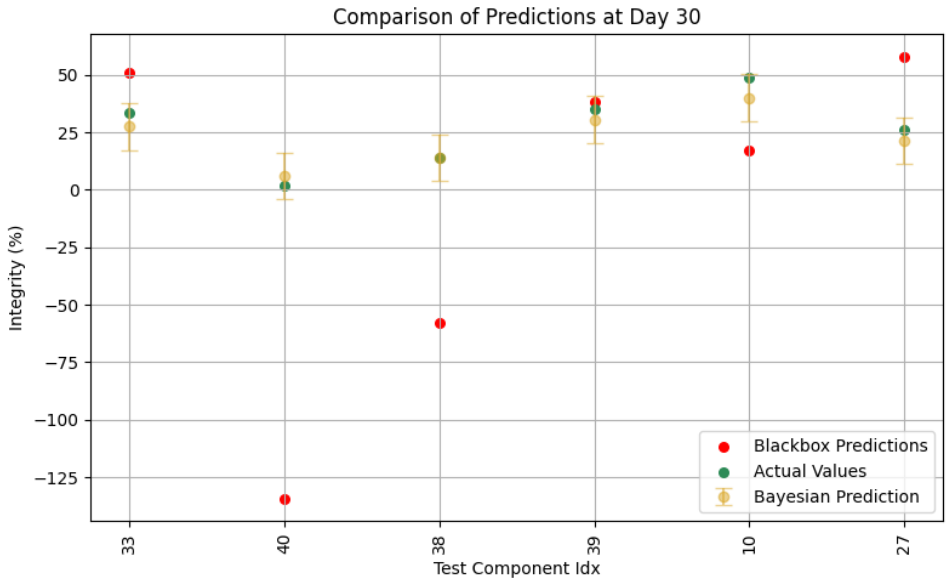


Figure 1.8: Black Box model performance.

The results show that the black box model performs worse than the enhanced model. While for some components, the black box model makes predictions that are close to the actual values, it produces errors, even predicting negative integrity values, which is not even physically possible. The enhanced model performs really well, with all of its predictions being close to the actual value, since it implements uncertainty through the posterior distributions. Thus this proves that the enhanced model is more reliable for the client's objectives.

1.4.2 Using Support Vector Machines

One engineering team member had asked why we do not use SVMs for this task, SVMs is a supervised learning tool for classification and regression, but they are point predictors, not probabilistic models. For this project, where quantifying uncertainty is required, SVMs will not perform as well as the Bayesian methods we used. Another reason is that Bayesian methods are better suited for tasks where there is not much data available, like in the case for components 50-74. Thus, Bayesian modelling is the most appropriate choice for this task.