Coursework 2c Report

John Steward

April 29, 2025

1 Introduction

We have been presented with a project that involves investigating the rate of degradation of different mechanical components. The data that has been presented to us includes 50 components with integrity measurements that are observed over a period of up to $t = 45$ days, and a further 25 components that have many fewer integrity measurements over just the first few days.

These components also each have their own characteristics, denoted by X1 to X5. These will be used in section 2.2 to help model the degradation rate of each component more accurately.

This kind of modelling can be difficult for traditional techniques to tackle since, even when looking at our training data, the data measurements are quite sparse and noisy, potentially leading to overfitting. Bayesian modelling is very useful in this aspect, being very effective at modelling functions while using very little data, being able to model any uncertainties using probability distributions that we will be able to sample from. Using this Bayesian hierarchical framework, we will be able to effectively model the degradation rate of the components in future timesteps where there have been no data points, up to $t = 60$ days. With this framework, we are also able to easily visualise how the parameters relate to each other, and the model as a whole, leading to an easily interpretable model, unlike black-box models which can be difficult to interpret.

In this report, We will discuss the two modelling methods for this task, these being: Our baseline model, that does not take into account the individual component characteristics, and our enhanced model, that takes these characteristics into account, leading to a more effective model. I will discuss the details of these models, and how they show where potential causes of degradation lie. I will then address the additional queries that our client has, being the use of a black-box model, and of Support Vector Machines (SVM). I will discuss the shortcomings of both of these modelling techniques for this task, and why our Bayesian model is more appropriate.

2 Predictive Modelling

2.1 Baseline Model

Our baseline model function is shown below as:

$$y_i(t) = f_i(t) + \epsilon$$

$$f_i(t) = u_i \exp\{-v_i t/100\}$$

where u is our intercept term, which is the value of f_i when $t = 0$, and v is our degradation rate term. These range from 80-100, and 0-10 respectively. Since there is a potential range in these values, using our Bayesian framework, we are able to model these parameters as probability distributions which we can sample from and create an overall distribution of function measurements, which we take the mean of for our final function.

Since we do not know much about these two parameters, we choose to model them using a uniform distribution from which we sample between the known values of 80-100 for u and 0-10 for v. This means that each value has an equal chance of occurring for each sample and, when we sample enough times and model our function, we can get a more representative model of these parameters and the degradation rate of the components. The sampled distributions for these parameters can be seen in figure 1.

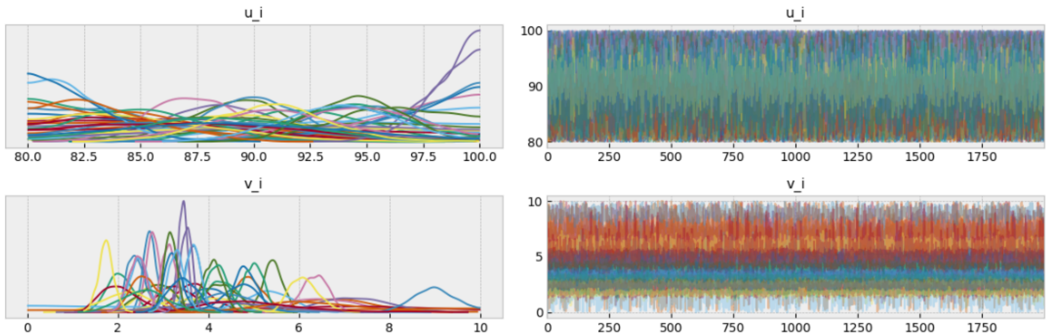

Figure 1: u and v distributions that we sample from for our baseline model

We also model the noise parameter σ, however for this parameter, we know that it is an approximate Gaussian distribution and, since we cannot have a negative noise, we sample from a half normal Gaussian distribution, which can be seen in figure 2.

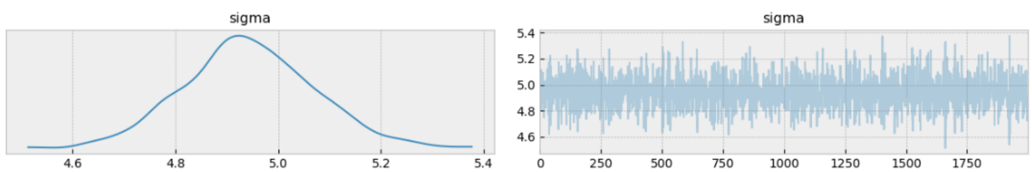

Figure 2: Sigma distribution that we sample from for our baseline model

Results from this model can be seen in figures 10, 11, 12, and 13, where figure 10 shows the result on one of our training datapoints, and the others are from our blind test set.

This model works as a good baseline for modelling a generic component, however it does present issues when it comes to modelling the specific components with their own characteristics. Since this model does not take these characteristics into account, the model assumes that all components have the same properties, which leads to more function uncertainty in the model, and potentially inaccurate predictions. To solve this issue, we look at our enhanced model.

2.2 Enhanced Model

This model is in the same format as the baseline model defined above, however $f_i(t)$ is defined as:

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

Where u and v are as defined above, with their sample distributions shown in figure 3, x_j from $1 \leq j \leq 5$ are the component characteristics for component i. With this enhancement to our baseline model, we can now gain more information from each individual component, which will lead to a more certain function when estimating the degradation for timesteps with no datapoints. For the weight parameters w, we sample from a Gaussian normal distribution, with a mean of 0 and standard deviation of 1. These sampled distributions can be seen in figure 4, and implications of these distributions will be discussed in section 2.2.1.

For this model, sigma is also sampled in the same way, using a Gaussian half normal distribution. (See figure 5).

When modelling the same components as shown in section 2.1, we can see that, for our training component, the function uncertainty is much lower for our enhanced model than for our baseline model (See figure 14). We have been able to gain much more insight from this relatively small amount of data, in order to model a more representative function of the degradation of this component.

When we look at the same blind test components (figures 15, 16, and 17), we can see that the function uncertainty is still smaller than with the baseline model, since we have been able to gain more

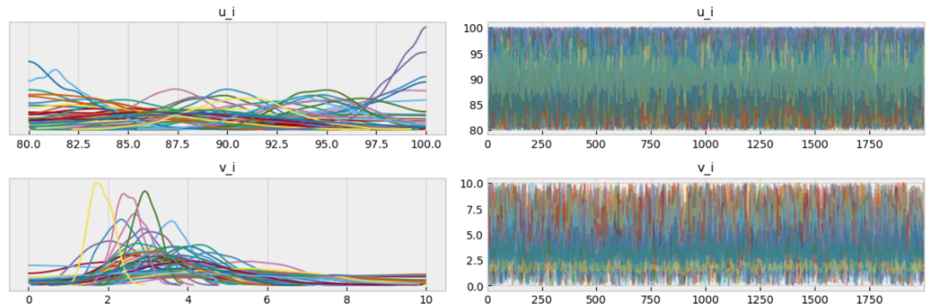

Figure 3: Sampled distributions of u and v for our enhanced model

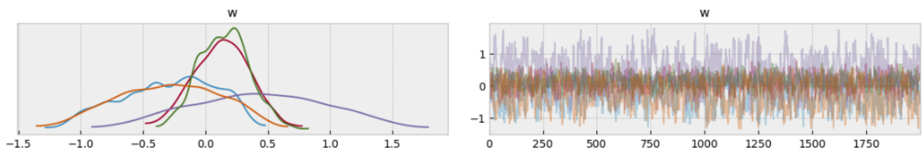

Figure 4: Plot trace of the weight parameter distributions for the component characteristics

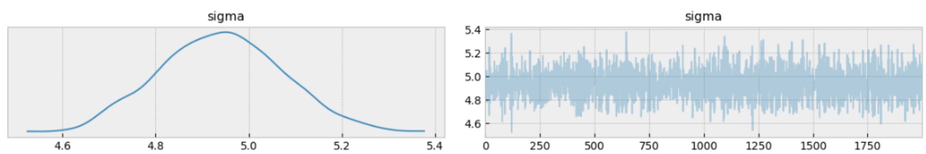

Figure 5: Sampled sigma distribution for our enhanced model

insight from the existing data, and model a more representative function of the individual component, rather than having to generalise to all components, without considering their individual characteristics.

2.2.1 Potential Causes of Degradation

When looking at our weight distributions from figure 4, we can see that two of the weight parameters have much higher peaks and tighter distributions than the others, these being the weights that correspond to X2 and X4. This shows that these two characteristics are more important when it comes to the degradation of components than the other characteristics, and are more informative in deciding whether the component will degrade more quickly or slowly. With the other weights being more spread out across the distribution, this means that they have less of an impact on the degradation of the components than X2 and X4, and will need to be taken less into consideration when analysing new components to see whether they are likely to degrade quickly. Without using our Bayesian hierarchical model, it would be much more difficult to be able to confidently come to this conclusion with such sparse data, however, since we are using our HMC sampler for all of these distributions, we are able to sample the full posterior for every parameter in order to gain a fuller understanding of each. For this reason, I would suggest to the client that they should analyse these two component characteristics in more detail, so that they can start to decrease the degradation rate of the components.

3 Additional Queries

3.1 Black-Box Model

To demonstrate the use of a black-box model to compare to our initial enhanced Bayesian hierarchical model, I implemented a simple neural network that uses a Mean Square Error loss function for training. I trained the same neural network twice, with the only changes being the number of epochs used to train the model. The first was trained over only 100 epochs, and the second was trained over 1000

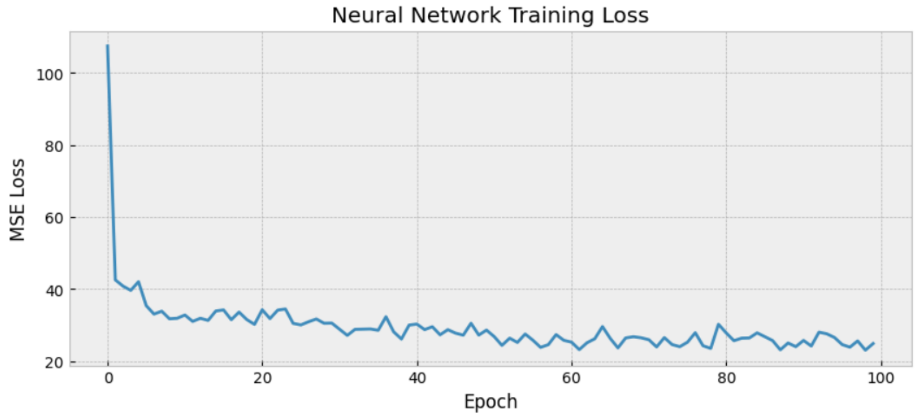

Figure 6: MSE Loss while training the Neural Network over 100 epochs

epochs. The graphs of the training errors are shown in figures 6 and 7 respectively. As shown in these figures, the MSE loss reduces extremely quickly, even when training with such few iterations.

Since there are so few datapoints, there is little room for the training algorithm to go wrong, so the loss is reduced extremely quickly. This does also, however, present a problem. As we can see from figure 8 and especially figure 9, the neural network model has massively overfit the data. This is again because there are so few datapoints, that it very quickly learns that they must be a true representation of the underlying function, when in fact that is not the case. This is where our Bayesian modelling shines, where we can model uncertain parameters probabilistically, allowing us to generalise to a function using very little data.

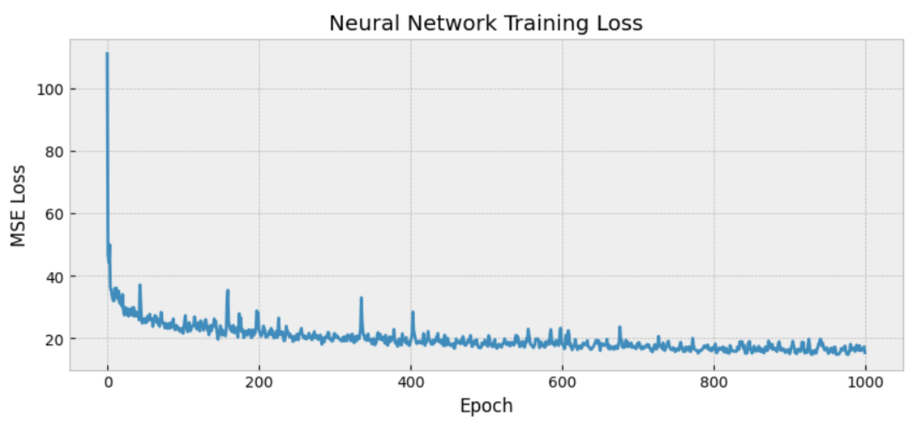

Figure 7: MSE Loss while training the Neural Network over 1000 epochs

3.2 Support Vector Machine

Support Vector Machines can be very powerful tools when it comes to classification, and regression. When used for a regression task, they fit a function that stays within a specified margin of error, being penalised for datapoints that are too far away.

This then presents the same issue as with the black-box model where, due to there being so few datapoints to train on, it can very quickly overfit to the training data, causing it to fall short when it comes to testing, when there are even fewer datapoints to model from.

We saw in section 3.1 that Bayesian modelling is much better equipped than an SVM for modelling functions using very few datapoints.

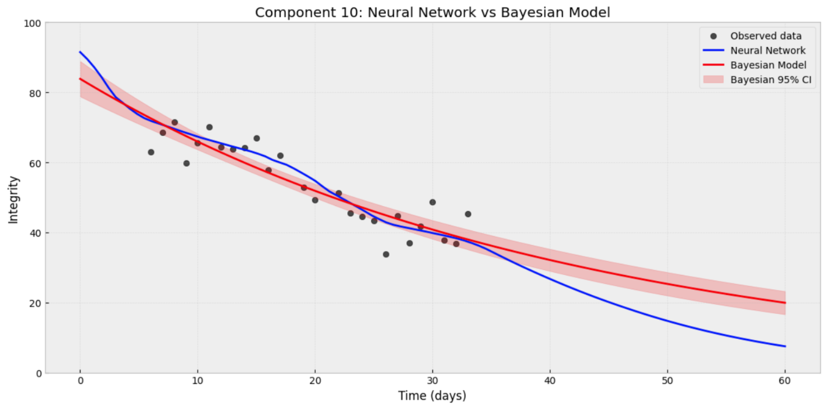

Figure 8: Comparison of the Neural Network trained over 100 epochs and our enhanced Bayesian model

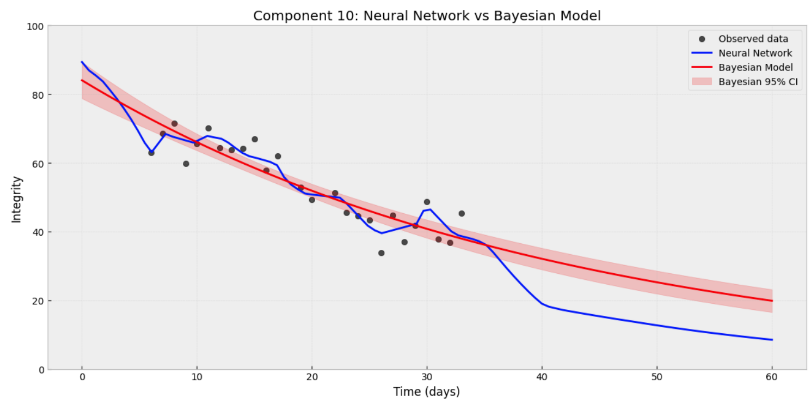

Figure 9: Comparison of the Neural Network trained over 1000 epochs and our enhanced Bayesian model

4 Conclusion

We have shown in this report that using a Bayesian hierarchical modelling framework is an extremely useful technique for modelling functions using very sparse data. This is thanks to the ability to model all of our priors as probability distributions, from which we can sample in order to explore the entire posterior, gaining a full insight into the relationships between the prior distributions and the overall model output function.

Using these techniques, we are also able to interpret our data much more easily than if we were to use a black-box model such as an Artificial Neural Network, as we did in section 3.1. For example, when we sample our weights for the component characteristics, we can see that the weights that correspond to X2 and X4 are more informative than the others, and we were able to suggest to the client that they analyse these in more detail to improve the component quality.

This technique is also useful in avoiding overfitting functions to the given data, and when we compare it to our black-box model shown in figures 8 and 9, we can see that we get a much smoother function when using our Bayesian model than we do with our neural network, showing that the neural network has overfit the data, even when using relatively few iterations to train. Overfitting is also a problem when it comes to using SVMs on very sparse data, which will minimise its loss just using the immediate data from those very few data points and fit a very tight function to them, instead of being able to sample from them and form a general function.

In conclusion, using a Bayesian hierarchical model is a very useful technique for projects such as these, with very low amounts of data, as we can sample prior distributions and explore the entire posterior to gain insight from the small amount of data available to us.

5 Appendix

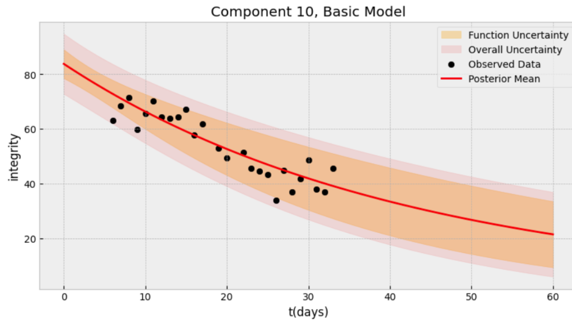

Figure 10: Our modelled function for component 10, from our baseline model

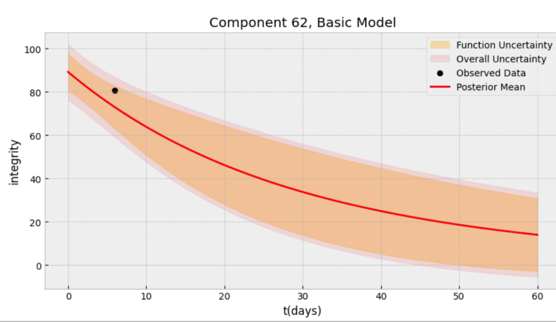

Figure 11: Our modelled function for component 62, from our baseline model

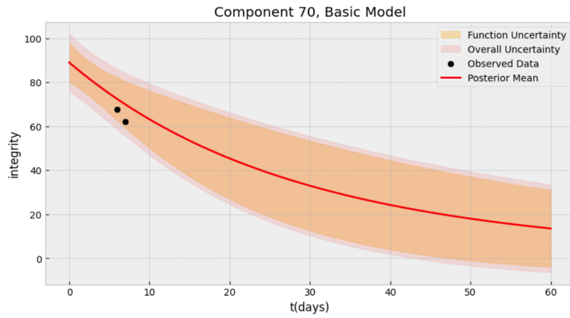

Figure 12: Our modelled function for component 70, from our baseline model

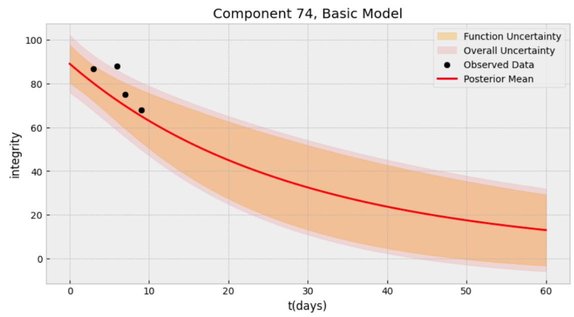

Figure 13: Our modelled function for component 74, from our baseline model

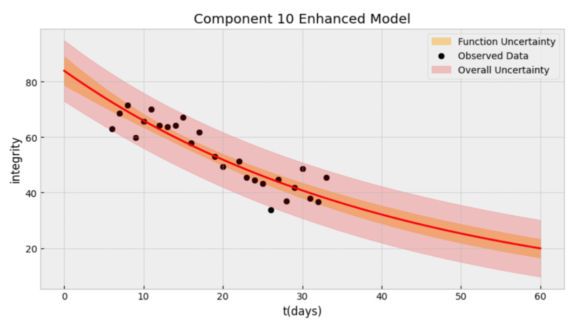

Figure 14: Our modelled function for component 10, from our enhanced model

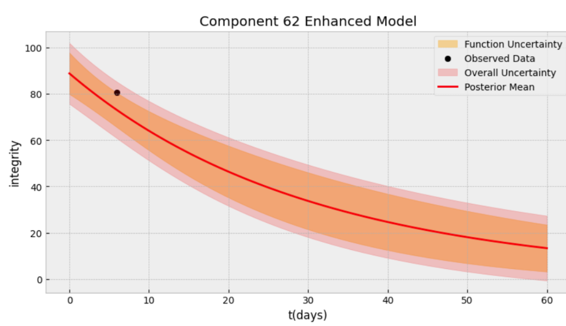

Figure 15: Our modelled function for component 62, from our enhanced model

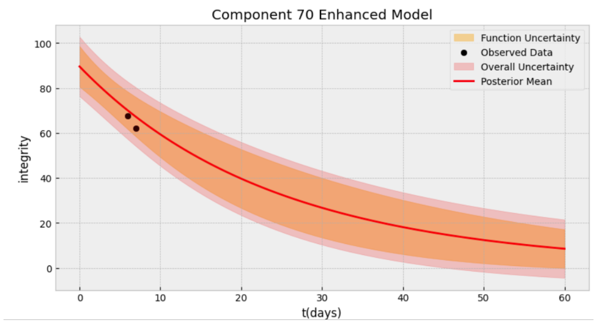

Figure 16: Our modelled function for component 70, from our enhanced model

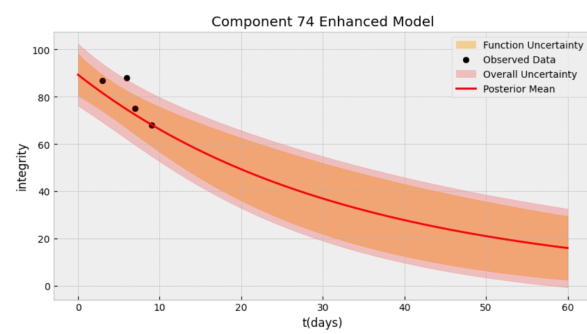

Figure 17: Our modelled function for component 74, from our enhanced model