Novelty Detection
with
Extreme Value Theory
in
Jet Engine Vibration Data

David A. Clifton
St. Cross College

Supervised by
Professor Lionel Tarassenko

Submitted: Trinity Term, 2009

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy
Novelty Detection with Extreme Value Theory in Jet Engine Vibration Data

David A Clifton
St. Cross College
Doctor of Philosophy
Trinity Term, 2009

ABSTRACT

Gas-turbine engines are critical to the operation of most industrial plants, aircraft, and heavy vehicles, and their associated maintenance costs can be high. Gas-turbine engine manufacturers typically adopt a condition monitoring approach, in which intelligent data analysis systems are employed to assess the “health” of engine components. However, existing methods generate large numbers of false alarms, such that they are typically ignored or deactivated. Furthermore, existing methods take a population-based approach, in which data acquired from the engine-under-test are compared to models constructed using data acquired from a population of similar engines. The variability between engines is such that existing alerting systems are insensitive to subtle engine events, and typically do not detect precursors to engine failure.

This thesis proposes an engine-specific methodology to overcome the problems described above, in which data acquired from an engine are compared to a probabilistic model constructed using data previously acquired from the same engine, rather than to a generic population-based model. This method is fully automatic, such that the characteristics of each engine are learned during its service life.

This thesis describes three complementary approaches to novelty detection, suitable for use in three different operational environments: (i) an approach suitable for high-bandwidth data, in which features are extracted from vibration spectra, their distribution modelled during periods of “normal” operation, which is then used to identify newly-acquired data that are “abnormal” with respect to that model; (ii) an approach for the analysis of very-high bandwidth data, in which entire vibration spectra are modelled, rather than features extracted from them; and (iii) an approach suited to low-bandwidth data, as typically occurs with on-wing engine health monitoring systems, in which engine vibration measurements observed during flights are summarised by a small number of data, which are then used for novelty detection. In each case, the proposed methods significantly outperform the monitoring systems implemented in existing engine health monitoring systems.

In addition to applying novelty detection to jet engine vibration data, the improvements to existing novelty detection theory that are proposed in this thesis are also applicable to data-modelling problems in general. We describe the limitations of existing methods of performing novelty detection, and show how techniques based on extreme value theory (EVT) can be used to overcome some of these limitations in a principled, probabilistic manner. We extend EVT (which is traditionally used to analyse the occurrence of rare events in univariate, unimodal data) for use with multivariate, multimodal data, as is required for analysing data from complex systems. We also extend EVT into a Bayesian framework, such that uncertainty in our model can be incorporated into the analysis. This reduces the false alarm rate to levels that allow the system to be used in practice.
Dedication

To Him from whom all just works do proceed,

and

我的小猫.
Acknowledgements

Many people played a part in the production of this thesis, and it is my great pleasure to thank them here. Their name is Legion, for they are many.

Much of the work described in this thesis (along with a large proportion more) arose directly from listening to interesting problems encountered by staff at Rolls-Royce PLC. In particular, thanks go to Dr. Steve King and Dennis King for their expertise in all things gas-turbine, and their boundless enthusiasm for new-fangled technology. Paul Anuzis and Robert Slater proved robust, yet encouraging, foils for my endless hours of presentations filled with equations. Thanks are due also to Dr. Mark Walters, David Ault, and the rest of their team for proving so helpful with the provision of data.

This work was sponsored by Oxford BioSignals Ltd., and I would like to thank Paul Nicholls in particular for his part in arranging the funding agreement, via the DTI-funded “HECToR” project. Thanks also to Drs. Nick McGrogan and Emma Braithwaite for their support and encouragement. I enjoyed (and continue to enjoy) many hours of discussion with Drs. Iain Strachan and Sriniv Sundaram. Thanks are also due to Ian Shaylor, Ken McDonald, Tom Collings, and Philip Elliot for implementing the work from this thesis in commercial-grade software, and for the resultant feedback that proved so very beneficial. Thanks to Mike Nichols of J.A. Kemp and Co. for collaborating on the patents related to this work, from which the clarity of much of the presentation of this thesis has benefited.

I was but the third muskateer during the three-year HECToR project to Drs. Peter Bannister and Simukai Utete, whom I wish to thank for their endless energy and guidance. I believe I am at least partially deaf in one ear due in part to driving from Oxford to Derby once per month in the ear-splitting music box which is Dr. Bannister’s 120 m.p.h. Subaru.

To the members of the BSP (née SPANNER) research group old and new, many thanks for making our environment the stimulating, enjoyable, and cake-laden place that it is. Special thanks are due to Dr. Alistair Hann, Dr. Mark Ebden, Dr. Nick Hughes, Dr. Vernon Bailey, and Dr. Oliver Gibson for all manner of useful discussions; to Dr. Thomas Brennan and Dr. Mel “Mark” Larsen for being sterling companions on the long (long) road to the DPhil; and to David Wong and Samuel Hugueny for being unreasonably talented, unfathomably interested, and for asking difficult questions. Thanks to Dr. Christina Orphanidou for being a fount of useful advice, and for providing encouragement at the frequency required (approx. $1.157 \times 10^{-5}$ Hz).

Very many thanks to Drs. Gari Clifford and Neil Townsend for their support in the nascent stages of this DPhil, without whom it may never have grown up at all.

Finally, there are two people for whom I must dedicate their own paragraphs. Though I could never offend his considerable modesty by offering such hyperbolic thanks in person, Prof. Lionel Tarassenko has taught me more than I could learn about matters academic and diplomatic. I would describe Lionel as an “old fashioned” supervisor, and those that know me will understand that I can offer no higher form of praise. Thanks also to Lionel for correcting my spellings and, grammar1.

By curious coincidence, I have the very great pleasure of thanking Dr. Lei Clifton both for her support with the novelty detection aspects of my work and, more importantly, for being a constant and loving wife throughout my degree.

To all of the above, and to those mistakenly unnamed, my most sincere thanks.

---

Related Publications

**A Comparison of Approaches to Multivariate EVT for Novelty Detection**

**Novelty Detection with Multivariate Extreme Value Theory, Part I**

**Novelty Detection with Multivariate Extreme Value Theory, Part II**

**Probabilistic Approaches to Condition Monitoring of Aerospace Engines**

**Novelty Detection**

**Novelty Detection in Jet Engine Vibration Spectra**

**Aircraft Engine Health Monitoring with Density Modelling and EVT**

**Automated Novelty Detection in Industrial Systems**

**A Framework for Novelty Detection in Jet Engine Vibration Data**

**Bayesian Extreme Value Statistics for Novelty Detection in Gas-Turbine Engines**

**High Dimensional Visualisation for Novelty Detection**

A Data Mining Approach to Reveal Patterns in Aircraft Engine and Operational Data

Condition Monitoring of Manufacturing Processes

Novelty Detection in Large-Vehicle Turbochargers

Specific and Generic Models for Jet Engine Novelty Detection

Trending Performance Parameters for Aircraft Condition Monitoring

Application of an Intuitive Novelty Metric for Jet Engine Condition Monitoring

Learning Shape for Jet Engine Novelty Detection

Related Patents
Method and Apparatus for Monitoring and Analyzing Vibrations in Rotary Machines

System Monitor

Novelty Detection in Gas-Turbine Engine Data

Machining Process Monitor
“My answer to the skeptics is that if people aren’t given well-founded methods like Extreme Value Theory, they’ll just use dubious ones instead.”

- Jonathan Tawn,
in Embrechts et al. (2008)
## Contents

1 **Introduction**  \hspace{1cm} 1
   1.1 The Need for Engine Health Monitoring  \hspace{1cm} 1
   1.2 Existing EHM Techniques  \hspace{1cm} 2
   1.3 Novelty Detection  \hspace{1cm} 5

2 **Jet Engine Vibration Data**  \hspace{1cm} 7
   2.1 Introduction  \hspace{1cm} 7
   2.2 Jet Engine Operation  \hspace{1cm} 7
   2.3 Pre-processing  \hspace{1cm} 9
   2.4 Feature Extraction  \hspace{1cm} 11
   2.5 Vibration Signatures  \hspace{1cm} 13
   2.6 Application of Novelty Detection  \hspace{1cm} 15
   2.7 Methodology  \hspace{1cm} 17
   2.8 Methodology for High- and Low-Bandwidth Methods  \hspace{1cm} 27

3 **Methods of Novelty Detection**  \hspace{1cm} 29
   3.1 Introduction  \hspace{1cm} 29
   3.2 Distance-Based Novelty Detection  \hspace{1cm} 29
   3.3 Probabilistic Novelty Detection  \hspace{1cm} 35
   3.4 Setting a Probabilistic Novelty Threshold  \hspace{1cm} 43
   3.5 Conclusions  \hspace{1cm} 46

4 **Extreme Value Theory**  \hspace{1cm} 48
   4.1 Classical Extreme Value Theory  \hspace{1cm} 48
   4.2 Setting Novelty Thresholds Using $P$  \hspace{1cm} 50
   4.3 Continuous Monitoring  \hspace{1cm} 52
   4.4 Existing EVT Approaches  \hspace{1cm} 54
   4.5 Conclusions  \hspace{1cm} 57

5 **High-Bandwidth Novelty Detection**  \hspace{1cm} 58
   5.1 Introduction  \hspace{1cm} 58
   5.2 Multivariate EVT  \hspace{1cm} 58
   5.3 EVT for Multimodal Models  \hspace{1cm} 71
   5.4 Understanding the EVD  \hspace{1cm} 78
   5.5 The $\Psi$-Transform  \hspace{1cm} 80
   5.6 Conclusions  \hspace{1cm} 86

6 **High-Bandwidth Novelty Detection with Tracked Order Data**  \hspace{1cm} 89
   6.1 Introduction  \hspace{1cm} 89
   6.2 Data  \hspace{1cm} 89
   6.3 Methodology  \hspace{1cm} 91
   6.4 Results  \hspace{1cm} 107
   6.5 Conclusion  \hspace{1cm} 128
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Engine tachometer pulses</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Engine vibration spectra</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Finding tracked order amplitude and frequency</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>An industry-standard speed-based vibration representation</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>A speed-based vibration signature</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Engine tachometer pulses</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>The conventional train-validate-test procedure</td>
<td>19</td>
</tr>
<tr>
<td>2.8</td>
<td>A confusion matrix</td>
<td>20</td>
</tr>
<tr>
<td>2.9</td>
<td>On-line training and testing with abnormal data</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>Train-validate-test for engine-specific novelty detection</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>An SVM decision boundary</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Integrating a multimodal probability distribution</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>Breaking $P$ thresholds</td>
<td>51</td>
</tr>
<tr>
<td>4.2</td>
<td>EVT for quantifying threshold exceedance</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>EVT with continuous monitoring</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Example cdfs and EVDs</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>Increasing classical EVT thresholds</td>
<td>54</td>
</tr>
<tr>
<td>4.6</td>
<td>Disadvantages of existing EVT novelty detection methods</td>
<td>56</td>
</tr>
<tr>
<td>5.1</td>
<td>Multivariate EVDs are hyperspherical Gumbel distributions</td>
<td>61</td>
</tr>
<tr>
<td>5.2</td>
<td>Fitting Gumbel distributions</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>EVT parameters for increasing dimensionality</td>
<td>66</td>
</tr>
<tr>
<td>5.4</td>
<td>EVDs for a bivariate Gaussian distribution</td>
<td>67</td>
</tr>
<tr>
<td>5.5</td>
<td>Fitting EVDs to the bivariate Gaussian distribution</td>
<td>68</td>
</tr>
<tr>
<td>5.6</td>
<td>Probability mass for increasing dimensionality</td>
<td>69</td>
</tr>
<tr>
<td>5.7</td>
<td>EVDs for a general bivariate Gaussian distribution</td>
<td>71</td>
</tr>
<tr>
<td>5.8</td>
<td>Multimodal $p(x)$</td>
<td>72</td>
</tr>
<tr>
<td>5.9</td>
<td>Disadvantages of the “winner-takes-all” EVD method</td>
<td>75</td>
</tr>
<tr>
<td>5.10</td>
<td>EVD for overlapping kernels</td>
<td>77</td>
</tr>
<tr>
<td>5.11</td>
<td>Mapping from $p(x)$ contours to the EVD</td>
<td>79</td>
</tr>
<tr>
<td>5.12</td>
<td>$Ψ$-transform of probabilities $p(x)$</td>
<td>80</td>
</tr>
<tr>
<td>5.13</td>
<td>Finding the EVD of a multimodal model of normality</td>
<td>82</td>
</tr>
<tr>
<td>5.14</td>
<td>EVD for a Parzen window model</td>
<td>83</td>
</tr>
<tr>
<td>5.15</td>
<td>Finding the EVD of high-dimensional models of normality</td>
<td>84</td>
</tr>
<tr>
<td>5.16</td>
<td>Visualising the EVD in 3-D</td>
<td>85</td>
</tr>
<tr>
<td>6.1</td>
<td>Training, validation, and test data for high-bandwidth analysis</td>
<td>91</td>
</tr>
<tr>
<td>6.2</td>
<td>Model construction process for high-bandwidth data</td>
<td>94</td>
</tr>
<tr>
<td>6.3</td>
<td>Dividing a time-series of engine speeds into speed bands</td>
<td>95</td>
</tr>
<tr>
<td>6.4</td>
<td>Proportion of data retained for varying speed-based rejection threshold</td>
<td>96</td>
</tr>
<tr>
<td>6.5</td>
<td>Example of speed-based data rejection</td>
<td>98</td>
</tr>
<tr>
<td>6.6</td>
<td>Number of data in speed bands before and after speed rejection</td>
<td>98</td>
</tr>
<tr>
<td>6.7</td>
<td>Architecture of a NeuroScale network</td>
<td>100</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.8</td>
<td>Visualisation of normalised data from datasets $A_{1,1} \ldots A_{1,3}$</td>
<td>102</td>
</tr>
<tr>
<td>6.9</td>
<td>Training error for 10 candidate cross-validation GMMs</td>
<td>104</td>
</tr>
<tr>
<td>6.10</td>
<td>Selecting model complexity using cross-validation</td>
<td>106</td>
</tr>
<tr>
<td>6.11</td>
<td>Negative log-likelihood for the best-fit GMM</td>
<td>106</td>
</tr>
<tr>
<td>6.12</td>
<td>Distribution of example model errors during training</td>
<td>107</td>
</tr>
<tr>
<td>6.13</td>
<td>Visualisation of data from dataset $A_1$</td>
<td>109</td>
</tr>
<tr>
<td>6.14</td>
<td>Normal data from dataset $A_{1,2}$</td>
<td>112</td>
</tr>
<tr>
<td>6.15</td>
<td>Abnormal data from dataset $A_{1,5}$</td>
<td>113</td>
</tr>
<tr>
<td>6.16</td>
<td>Abnormal data from dataset $A_{1,7}$</td>
<td>114</td>
</tr>
<tr>
<td>6.17</td>
<td>Abnormal data from dataset $A_{1,8}$</td>
<td>115</td>
</tr>
<tr>
<td>6.18</td>
<td>Visualisation of dataset $A_2$</td>
<td>116</td>
</tr>
<tr>
<td>6.19</td>
<td>Normal data from dataset $A_{2,2}$</td>
<td>118</td>
</tr>
<tr>
<td>6.20</td>
<td>Abnormal data from dataset $A_{2,10}$</td>
<td>119</td>
</tr>
<tr>
<td>6.21</td>
<td>Classification results from dataset $A_1 \ldots A_2$</td>
<td>123</td>
</tr>
<tr>
<td>6.22</td>
<td>Abnormal data from dataset $A_{3,6}$</td>
<td>125</td>
</tr>
<tr>
<td>6.23</td>
<td>Abnormal data from dataset $A_{4,5}$</td>
<td>126</td>
</tr>
<tr>
<td>6.24</td>
<td>Abnormal data from dataset $A_{4,6}$</td>
<td>127</td>
</tr>
<tr>
<td>7.1</td>
<td>Partition of datasets for spectral analysis</td>
<td>132</td>
</tr>
<tr>
<td>7.2</td>
<td>Dividing speed-based spectra into bins</td>
<td>134</td>
</tr>
<tr>
<td>7.3</td>
<td>Speed-quantisation effects</td>
<td>135</td>
</tr>
<tr>
<td>7.4</td>
<td>Noise log amplitude-frequency relationship</td>
<td>136</td>
</tr>
<tr>
<td>7.5</td>
<td>Examining the contents of speed-frequency bins</td>
<td>138</td>
</tr>
<tr>
<td>7.6</td>
<td>Tracked orders and noise in a speed bin</td>
<td>138</td>
</tr>
<tr>
<td>7.7</td>
<td>Bin populations for spectral data representation</td>
<td>139</td>
</tr>
<tr>
<td>7.8</td>
<td>Fitting gamma distributions to “trimmed” datasets</td>
<td>140</td>
</tr>
<tr>
<td>7.9</td>
<td>Gamma mean and standard deviation for speed-frequency windows</td>
<td>141</td>
</tr>
<tr>
<td>7.10</td>
<td>Classical EVD and cumulative EVD for a gamma distribution</td>
<td>142</td>
</tr>
<tr>
<td>7.11</td>
<td>Classical and ML Gumbel distributions for a gamma distribution</td>
<td>143</td>
</tr>
<tr>
<td>7.12</td>
<td>Noise-floor estimates using classical and ML EVT methods</td>
<td>143</td>
</tr>
<tr>
<td>7.13</td>
<td>Vibration amplitudes for selected speed bins</td>
<td>144</td>
</tr>
<tr>
<td>7.14</td>
<td>Computing $P_e(x)$ and novelty scores $z(x)$ for test spectra</td>
<td>146</td>
</tr>
<tr>
<td>7.15</td>
<td>Comparing vibration amplitude, $P_e(x)$ and $z(x)$</td>
<td>148</td>
</tr>
<tr>
<td>7.16</td>
<td>$C$ matrices characterising training spectra</td>
<td>151</td>
</tr>
<tr>
<td>7.17</td>
<td>Proportion of speed-frequency bins masked, with varying $N_m$</td>
<td>152</td>
</tr>
<tr>
<td>7.18</td>
<td>Spectral data from dataset $B_1$</td>
<td>154</td>
</tr>
<tr>
<td>7.19</td>
<td>Proportion of test data classified “abnormal” for dataset $B_1$</td>
<td>155</td>
</tr>
<tr>
<td>7.20</td>
<td>Varying $N_m$ for dataset $B_1$</td>
<td>156</td>
</tr>
<tr>
<td>7.21</td>
<td>$P_e(x)$ for NTOs from dataset $B_1$</td>
<td>157</td>
</tr>
<tr>
<td>7.22</td>
<td>$P_e(x)$ for normal data from dataset $B_1$</td>
<td>157</td>
</tr>
<tr>
<td>7.23</td>
<td>$P_e(x)$ for NTOs from dataset $B_2$</td>
<td>158</td>
</tr>
<tr>
<td>8.1</td>
<td>A speed-based tracked order signature</td>
<td>162</td>
</tr>
<tr>
<td>8.2</td>
<td>Setting novelty thresholds with maxima</td>
<td>164</td>
</tr>
<tr>
<td>8.3</td>
<td>Mean and standard deviation for increasing $N$</td>
<td>167</td>
</tr>
<tr>
<td>8.4</td>
<td>Conjugate prior distributions</td>
<td>173</td>
</tr>
<tr>
<td>8.5</td>
<td>Example of Bayesian recursive learning</td>
<td>174</td>
</tr>
<tr>
<td>8.6</td>
<td>Selected distributions for Bayesian learning</td>
<td>174</td>
</tr>
<tr>
<td>8.7</td>
<td>Sampling the posterior distribution</td>
<td>176</td>
</tr>
<tr>
<td>8.8</td>
<td>Sampling the $\mu$ distribution</td>
<td>178</td>
</tr>
<tr>
<td>8.9</td>
<td>Sampling the distribution of the precision</td>
<td>179</td>
</tr>
<tr>
<td>8.10</td>
<td>Sampling the $\lambda$ distribution</td>
<td>180</td>
</tr>
<tr>
<td>8.11</td>
<td>Comparing B-EVT and ML-EVT</td>
<td>182</td>
</tr>
<tr>
<td>8.12</td>
<td>B-EVT and ML-EVT converging</td>
<td>182</td>
</tr>
<tr>
<td>8.13</td>
<td>B-EVT and ML-EVT thresholds</td>
<td>184</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>9.1</td>
<td>Setting novelty thresholds with the heuristic method</td>
<td>187</td>
</tr>
<tr>
<td>9.2</td>
<td>Example maxima distributions</td>
<td>189</td>
</tr>
<tr>
<td>9.3</td>
<td>B-EVT thresholds for varying $P_e(x)$</td>
<td>193</td>
</tr>
<tr>
<td>9.4</td>
<td>Model training for low-bandwidth applications</td>
<td>193</td>
</tr>
<tr>
<td>9.5</td>
<td>B-EVT thresholds for varying $P_e(x)$ with $C_1$</td>
<td>196</td>
</tr>
<tr>
<td>9.6</td>
<td>B-EVT thresholds using different prior distributions</td>
<td>198</td>
</tr>
<tr>
<td>9.7</td>
<td>B-EVT and other methods with $C_1$</td>
<td>199</td>
</tr>
<tr>
<td>9.8</td>
<td>B-EVT thresholds over time for $C_1$</td>
<td>200</td>
</tr>
<tr>
<td>9.9</td>
<td>Novelty thresholds for a “normal” speed bin in $C_1$</td>
<td>200</td>
</tr>
<tr>
<td>9.10</td>
<td>B-EVT thresholds for varying $P_e(x)$ with $C_2$</td>
<td>202</td>
</tr>
<tr>
<td>9.11</td>
<td>B-EVT false-positive classifications for $C_2$</td>
<td>203</td>
</tr>
<tr>
<td>9.12</td>
<td>B-EVT and other methods with $C_3$</td>
<td>205</td>
</tr>
<tr>
<td>9.13</td>
<td>B-EVT and other methods with $C_4$</td>
<td>207</td>
</tr>
</tbody>
</table>
List of Tables

6.1 Datasets used for high-bandwidth novelty detection ................. 89
6.2 Features available for high-bandwidth novelty detection ............. 92
6.3 Results of M-EVT with first test dataset .................................. 124
6.4 Results of M-EVT with second test dataset .............................. 128

7.1 Datasets used for novelty detection with full spectral data ........... 131
7.2 “Novel” classifications for dataset $B_2$ ..................................... 158
7.3 “Novel” classifications for dataset $B_3$ ..................................... 159

8.1 Total probability mass for Bayesian EVT with increased sampling .... 179
9.1 Datasets used for novelty detection with Bayesian EVT ............... 188
9.2 Results of applying the K-S test for Gaussianity ......................... 191
9.3 Data rejected by the K-S test for Gaussianity ............................. 191
9.4 Labels for training/validation datasets ..................................... 195
9.5 Classification results for low-bandwidth novelty detection .......... 195
9.6 Classification results for test datasets .................................... 204
9.7 Classification results after training and validation ...................... 204
Chapter 1

Introduction

1.1 The Need for Engine Health Monitoring

Gas-turbine engines are critical to the operation of most industrial plants, aircraft, and heavy vehicles such as military armour and transport ships, and their associated maintenance costs can be high. Traditionally, operators of gas-turbine engines have attempted to reduce these costs by performing preventative maintenance actions at fixed intervals, in an attempt to avoid potential engine failure. More recently, gas-turbine engine manufacturers have been adopting a condition monitoring approach instead, in which intelligent data analysis systems are employed to assess the “health” of engine components. The engine’s maintenance needs are determined according to its operating condition, rather than maintenance being performed at fixed periods of time.

These engine health monitoring (EHM) systems typically process data from engine-mounted sensors. Early warning of potentially hazardous engine conditions may result, the hope being that precursors of component failure may be identified in advance of actual failure. This is a prognostic approach to condition monitoring, and is useful for types of faults that may be prevented if identified soon enough. Faults for which there are no such precursors (e.g., a “bird strike”, in which a bird impacts an engine fan) require a diagnostic approach. Such systems automatically identify engine faults that have occurred, and may recommend restorative maintenance actions appropriate to the type of fault.

Gas-turbine engine manufacturers are increasingly offering a “service-based” approach to marketing their products, in which their customers are guaranteed availability of the purchased engine for an agreed percentage of the time (typically, 99% of the time).
To achieve this, manufacturers take on the responsibilities of engine condition monitoring, by embedding health monitoring systems within each engine unit and prompting preventative maintenance actions when necessary.

Condition monitoring techniques are also applied to engines during the development process, and throughout product testing. During development, engines are typically re-built and tested many times. Assessment of component health during this process can avoid component damage and potential hazards.

1.2 Existing EHM Techniques

1.2.1 Standard methods

Guidelines for the production of condition monitoring systems applied to aircraft gas-turbine engines have been available for many years (SAE International 1981, SAE International 1986, SAE International 1994). Engine monitoring is performed using either “on-line” systems, mounted within the aircraft, that perform analysis of engine data during flight, or “off-line” ground-based systems, to which engine data is downloaded from the aircraft at the end of a flight.

On-line condition monitoring systems are subject to strict certification requirements from international aviation authorities. The cost of producing systems to these required standards is very high, primarily due to the technical challenge of suppressing the large numbers of false alarms generated by many condition monitoring algorithms (Tumer et al. 1999). Methods currently used in on-line condition monitoring are relatively simple due to these constraints.

Off-line condition monitoring systems are limited by the amount of engine data that may be transferred from an aircraft. Typically, data storage and transfer costs are high, which constrains the quality and quantity of data available to ground-based monitoring systems in commercial use.

Standard methods of engine condition monitoring involve comparison of recorded engine parameters with fixed operational limits set by engine manufacturers (SAE International 1994), or simple univariate analyses in which trends in engine data are monitored in order to detect sudden shifts in operational behaviour, often indicators of potential failure (HereWatch 2003).
Current research on the condition monitoring of aircraft engines is focussed on providing more sophisticated methods of reliably determining engine condition and resulting maintenance recommendations, within the constraints of on-line and off-line monitoring systems. The following sections describe the general areas of current research.

1.2.2 Fault diagnosis using expert systems

Expert systems aim to capture the knowledge of domain experts, typically from engine manufacturers, in order to diagnose faults from engine data and recommend appropriate maintenance actions. A typical industry-standard software package commercially available for this purpose is the TEXMAS system (Collinge & Schoff 1988).

The accuracy of diagnosis may be improved through the construction of ever-larger databases of fault information for gas-turbine engines (Brown et al. 1994, Trammel et al. 1997). CASE-based reasoning (CBR) uses the intuition that new problems are likely to be similar to previously encountered problems of a similar nature. In this approach, a user-friendly software environment is utilised to capture expert knowledge and manage fault diagnoses. This can then be used to provide coordinated construction of expert-system databases for monitoring engines in civil aircraft between many different sites (Ong et al. 2004), using a framework for distributed Internet-based data analysis (Foster et al. 2002, Jackson et al. 2003). CBR has also been applied to the management of the maintenance process for gas-turbine engines in military aircraft (Abu-Hakima et al. 1989, Halasz et al. 1991).

However, the CBR approach is limited by the variation in operational behaviour between classes of aircraft engine, and even variation between engines of the same class, which can render the use of general fault databases ineffective when diagnosing engine-specific faults.

To avoid costs associated with reconstructing expert system databases for use with individual engines, the automated generation of the logical rules in the expert-system databases using “soft computing” techniques has been proposed.

“Fuzzy” expert systems consisting of fuzzy logic rules for characterising engine-specific faults have been combined with data-mining techniques (Fayyad et al. 1996)

\[1\]“Soft computing” methods allow reasoning in the presence of imprecision and uncertainty, often using probabilistic or fuzzy-logical frameworks.
in which rules are formulated from trends in the value of measured engine parameters (Yu et al. 2003).

Within a conventional expert system, neural network methods have been used to match features detected in performance parameter signals with diagnosis rules (DePold & Gass 1994, Dietz et al. 1988). These methods have also been applied to fault diagnosis of space shuttle engines (Powers et al. 1993), and to military armour (Greitzer et al. 1999). Here, diagnoses are created by combining sets of rules (comparing sensor output to fixed operational thresholds) and the outputs of a neural network classifying engine operation into one of several states, such as “idle”, “full power”, etc.

The combination of expert system databases from multiple engines of the same class has been attempted, aiming to generate systems that might generalise to other engines of that class (Pena et al. 2000).

1.2.3 Thermodynamical and frequency-based modelling

Thermodynamical models of gas-turbine engines have been used for condition monitoring by comparison of a model derived from actual engine data (usually temperatures and pressures) with a model constructed from the theoretical relationship between those engine data (Simani 2005). The residual error between the two models may be used to identify possible engine failures (Merrington 1994).

Such methods combine a theoretical approach to thermodynamical model construction with knowledge derived from engine testing (Polycarpou 1994). Models derived using finite-element analysis have been used in the condition monitoring of helicopter engines (Ganguli et al. 1997). Linear flow models have been constructed from fuel-flow data and applied to condition monitoring of the engine combustion system (Lieuwen 1999).

Frequency-domain approaches to modelling engine parameters have been used in fuel-flow monitoring (Evans et al. 1998), exhaust gas monitoring (Subbu et al. 2005), and more general diagnosis of engine faults (Winston et al. 1995, Ho et al. 1998). These models have been combined with auto-regressive, moving-average (ARMA) models for fuel-flow monitoring (Chiras et al. 2001), and for analysis of engines used in industrial plants (Booth et al. 2001).
1.2.4 Machine learning approaches

Determining engine condition by comparison of engine data with a model of normal behaviour may also be achieved using “soft computing” methods of model creation, such as machine learning or statistical pattern recognition techniques.

These methods offer the advantage that engine condition may be assessed without detailed theoretical system models. These data-driven approaches can be used to construct models of engine behaviour directly from engine data (Duda et al. 2001), in contrast to rule-based approaches that encode expert knowledge, and may be adapted for use with individual engine units, providing engine-specific condition monitoring. However, these methods can be sensitive to the quality and quantity of example data from which models of engine operation are constructed.

Artificial neural networks have been used to identify abnormalities in the operation of gas-turbine engines within industrial plants (Tarassenko et al. 1999, Tarassenko et al. 2000), in space shuttle engines (Meyer et al. 1994), and in engines used by heavy ships (Illi et al. 1994).

The support vector machine (SVM) method, a machine learning approach (Vapnik 1998) considered further in Chapter 3, has been used to identify example failures in aircraft engine production tests (Hayton et al. 2000), and in water-pump gas-turbine engines (Tax et al. 1999). Other methods of constructing models of engine behaviour during production tests using data-driven techniques have been successfully demonstrated using data from military aircraft engines (Nairac et al. 1997, Nairac et al. 1999).

Automated extraction of patterns describing normal and abnormal engine operating conditions with vibration data has been investigated (Austin et al. 2005), set within a general fuzzy-neural analysis framework (Austin et al. 1995).

1.3 Novelty Detection

Due to the high complexity of modern gas-turbine engines, detailed understanding of the relationships between the large numbers of engine components can be limited. Holistic models of system behaviour, such as models of engine vibration, are typically of low order and cannot provide accurate estimation of system behaviour under all operating conditions. Thus, conventional model-based monitoring systems can be unsuitable for
condition monitoring.

Furthermore, such high system complexity results in a large number of possible failure modes, the effects of which on observed data are often poorly defined. This can limit the effectiveness of conventional fault-specific failure detection schemes, which are trained to identify known, well-understood modes of failure.

This limitation of conventional techniques can be further compounded by the rarity of examples of abnormal behaviour in high-integrity systems, which are few in comparison to the quantity of examples of normal behaviour. Typically, there are insufficient examples of failure to construct accurate fault-detection systems.

Thus, in order to facilitate early warning of this large set of potentially ill-defined possible failures, the work described by this thesis takes a novelty detection approach. A model of normality is constructed from “normal” engine data, departures from which are classified as “abnormal” events.
Chapter 2

Jet Engine Vibration Data

2.1 Introduction

This chapter provides an introduction to jet engine operation, and to the acquisition of vibration data using transducers mounted on the engine casing. We then consider how vibration characteristics, or features, may be extracted from these transducers to build models of normality as a precursor to novelty detection.

Finally, this chapter describes the application of novelty detection in various scenarios, and discusses the problems that need to be overcome for each scenario. Principled methods of performing novelty detection are introduced, and extended for use with the different problems investigated in this thesis.

2.2 Jet Engine Operation

This thesis is primarily concerned with the condition monitoring of gas-turbine engines within the aerospace industry. These engines typically consist of a compressor section at the front of the engine that draws in air, pressurises it by up to forty times atmospheric pressure, and then delivers it to a combustion chamber at the rear. Fuel and compressed air are mixed and ignited within the combustion chamber, where the rapidly-expanding heated gas directed out of the rear of the engine provides motive force. Rear-mounted turbines extract energy from these exhaust gases to drive the compressor section, to which they are connected via a shaft.
2.2.1 Stages of compression

Modern aerospace engines divide the task of air compression from atmospheric pressure to that ultimately required within the combustion chamber into several stages. Many jet engines within the civil aerospace market involve three consecutive compression stages (Rolls-Royce PLC 2005): the low pressure (LP), intermediate pressure (IP), and high pressure (HP) stages. Air passes through each stage as it travels from the front of the engine to the rear, being further compressed by each, until it reaches the combustion chamber.

Each of the compressor stages is driven by its own turbine assembly, resulting in three corresponding turbine units situated within the exhaust stream at the rear of the engine. Each compressor is linked to its corresponding turbine by a separate shaft, all the shafts being mounted concentrically. In engines with three stages of compression, these are named the LP shaft, the IP shaft, and the HP shaft\(^1\).

2.2.2 Engine vibration and speed measurement

Vibration transducers are mounted at various points of the engine assembly, typically embedded within the external casing. These transducers often operate at different sampling frequencies \(f_s\) to allow varying resolutions of frequency analysis to take place, as will be later described.

The detection of departure from normal vibration behaviour requires knowledge of shaft rotational speed. Tachometers are mounted on the engine shafts for the measurement of shaft rotation. These consist of an optical or electromagnetic transducer monitoring a “comb” of \(N_t\) teeth that rotates with the shaft. Rotation of the engine shaft causes the tachometer to produce a train of square waves, whereby \(N_t\) square waves are observed for each complete shaft rotation.

An example is given in Figure 2.1(a), which shows 19 square waves in a period of 0.04s. Time-differencing the signal results in a pulse train, as shown in Figure 2.1(b), from which pulses of amplitude above some threshold \(h_\omega\) are detected. In the exemplar shown in the figure, a threshold of \(h_\omega \approx 0.5 \times 10^4\) is sufficient for detection of the pulses.

The LP shaft shown in this example has a tachometer comb of \(N_t = 60\) teeth, and

\(^1\)Military aerospace engines often include only LP and HP shafts. This thesis considers novelty detection in both two- and three-shaft engines.
so the rotational frequency is approximately $\frac{19}{(60 \times 0.04s)} = 7.91$ Hz. Rotational speed is usually expressed in terms of the percentage of maximum shaft speed. For the exemplar engine, 100% rotational speed of the LP shaft corresponds to 2,900 RPM, or a rotational frequency of 48.33 Hz. The tachometer pulses shown in the figure therefore correspond to rotational speed $\omega_{LP} = \left(\frac{7.91 \text{ Hz}}{48.33 \text{ Hz}}\right) \times 100\% = 16.4\%$.

The estimate of shaft rotation speed should not be sensitive to missing occasional pulses, and so in practice it is assumed that changes in rotational speed are negligible over some longer period, such as $t_\omega = 0.2s$. Rotational speed is then determined by counting the number of tachometer pulses that occur within successive time intervals of duration $t_\omega$.

### 2.3 Pre-processing

Pre-processing is invariably required before any vibration dataset can be analysed, as the analysis of raw transducer signals usually yields poor results. The main issues to
consider are described in the following subsections.

2.3.1 Data quality

The acquisition of data for building models of normality prior to novelty detection is susceptible to noise, both from the environment in which the engine is operating, and from the characteristics of the transducers for data acquisition. Noise-reduction techniques are therefore applied as a pre-processing step. Typically, this involves filtering the acquired vibration data according to prior knowledge (or estimation) of the noise process.

In this thesis, the analogue signals from the vibration transducers are initially low-pass filtered to remove high-frequency noise. Anti-aliasing filtering is then applied to the time-domain waveforms to prevent errors in digitisation.

2.3.2 Suitability of the data domain

The domain in which the data are acquired may not be best suited for characterising differences between “normal” and “abnormal” behaviour of the engine. Transforming the data into a new domain may allow more effective novelty detection.

For example, the engine vibration data considered by this thesis will exhibit peaks in vibration amplitude at the fundamental rotational frequency of the engine shafts. These vibration amplitudes can be used, as will be described later, to characterise the condition of the engine. The vibration characteristics are not easy to identify in the time domain and so the data acquired from the vibration transducers are transformed into the frequency domain using fast Fourier transforms (FFTs), using a system (Hayton et al. 2003) that computes FFTs every 0.2 s for each vibration transducer. Non-overlapping windows of 1,024 data points are used to compute 512 spectral components (or “bins”) across the range of frequencies $f = [0 \ f_s/2]$, for a transducer sampling frequency $f_s$. Engine vibration is assumed to be pseudo-stationary over this measurement period such that the resultant FFTs are assumed to be close approximations of the engine vibration spectra.
Figure 2.2: Vibration spectra for an example three-shaft engine. Fundamental tracked orders 1LP, 1IP, and 1HP are labelled, along with harmonic tracked orders 2HP and 3HP. Vibration amplitude is plotted in shades of grey, with darker shades corresponding to higher vibration amplitudes.

2.4 Feature Extraction

Prior knowledge of the system for which novelty detection is to be performed drives the choice of features. For example, section 2.3.2 noted that if an engine shaft rotates at $f_1$ Hz, a peak in vibration energy will occur at $f_1$ Hz, with corresponding harmonics occurring at multiples of $f_1$ Hz. Engine manufacturers use the values of vibration amplitude at the fundamental frequency and its harmonics, termed tracked orders (Nairac et al. 1997), as features deemed suitable for characterising the condition of the engine.

The tracked order corresponding to the fundamental rotational frequency $f_1$ of the LP shaft is termed 1LP, and its harmonics are termed 2LP, 3LP, 4LP, etc. Harmonics also occur at 0.5 $f_1$ and at 1.5 $f_1$. Similarly, tracked orders for the IP and HP shafts are termed 1IP, 1HP, 2IP, 2HP, etc.

Figure 2.2 shows vibration spectra for an example three-shaft engine. Tracked orders appear as peaks in vibration amplitude that persist through time, and vary in frequency according to the rotational frequency of each shaft. Note that in the example shown in
Algorithm 2.1 Find the amplitude $|Y|$ and frequency $f_Y$ of tracked order $Y$ in a vibration spectrum $X$ of $N_{\text{FFT}}$ components

**Ensure:** $N$ is the order of $Y$ (e.g., if $Y$ is 2LP, $N = 2$)

**Ensure:** $f$ is the rotational frequency of the shaft associated with $Y$

(i) Find spectral component $\hat{B} \in [1 \ldots N_{\text{FFT}}]$ that is closest to frequency $N \times f$

(ii) Search either side of spectral component $\hat{B}$ for a peak in vibration amplitude:

$y \leftarrow 0, \quad B \leftarrow 0$

for all $n$ such that $\hat{B} - 3 \leq n \leq \hat{B} + 3$ do

if $|X_n| > y$ then

$y \leftarrow |X_n|$ \hspace{1cm} (2.1)
$B \leftarrow n$ \hspace{1cm} (2.2)

end if

end for

(iii) Fit quadratic polynomial function $G(f)$ to the triplet $|X_{B-1}|, |X_B|, |X_{B+1}|$

(iv) Tracked order amplitude and frequency,

$|Y| = \max G(f)$ \hspace{1cm} (2.3)
$f_Y = \arg\max_f G(f)$ \hspace{1cm} (2.4)

the figure, the engine is slowly accelerating after $t > 50$ s, and so the frequencies at which the various tracked orders occur are slowly increasing. As the shafts are not mechanically connected in most modern engines\(^2\), there is a non-linear relationship between the shaft rotational speeds.

The $N_{\text{FFT}}$ spectral components may be summarised by a low number of tracked order vibration amplitudes, determined from vibration spectra by using Algorithm 2.1.

As a result of inaccuracy in the estimation $\hat{f}$ of the rotational frequency of engine shafts, it cannot be assumed that the frequency at which a peak in vibration amplitude occurs due to a tracked order is an exact multiple of $\hat{f}$. Thus, Algorithm 2.1 searches within a range of spectral components for that peak in vibration amplitude, $|Y|$, corresponding to the tracked order. The frequency resolution of the vibration spectrum is limited by the number of spectral components used in the FFT computation, $N_{\text{FFT}}$, and

\(^2\)The shafts are mounted concentrically, and aligned using free-rolling bearings.
so the algorithm interpolates between the peak in vibration amplitude corresponding to the tracked order, and the vibration amplitudes of neighbouring spectral components.

Figure 2.3 shows an example in which peaks in vibration amplitude corresponding to tracked orders occur at spectral components $b = 9$ and $b = 16$, and where the amplitude of other spectral components is due to noise. Following Algorithm 2.1, a three-point quadratic interpolation is shown between the peak vibration amplitude at $b = 9$ and the amplitudes of its neighbouring components. The vibration amplitude of this tracked order is thus estimated to be $|Y| = 0.052$ in s$^{-1}$, occurring at frequency $f_Y = 171.9$ Hz, which is between the frequencies of the spectral components at $b = 8$ and $b = 9$.

### 2.5 Vibration Signatures

Using the feature extraction process described above, a time-series of tracked order vibration amplitudes can be obtained, with a data rate $f_s$ equal to the rate at which FFTs are computed. For the system used to acquire data for this thesis, $f_s = 5$ Hz.

An industry-standard method for displaying and analysing vibration information is the “ZMOD”, which is a speed-based representation. An example is shown in Figure 2.4. Engine vibration is conventionally measured across the speed range $\omega = [0\%\ 100\%]$.
maximum speed of the corresponding shaft. This speed range is subdivided into $N_\omega$ equal bins. For the example shown in the figure, $N_\omega = 400$, and each bin corresponds to a 0.25% sub-range of the maximum shaft speed. Within each bin $b = 1 \ldots N_\omega$, all vibration amplitudes observed at that speed throughout a flight are stored (with the highest vibration amplitude stored in each bin plotted in the figure). The figure shows that tracked order vibration amplitudes appear as lines of high vibration amplitude throughout the speed-range, increasing in frequency with increasing shaft speed.

In this thesis, we will mostly consider the use of tracked order vibration amplitudes, rather than the full vibration spectra shown in the ZMOD representation, and so we plot the distribution of vibration amplitudes of individual tracked orders at each shaft speed. Such a speed-based representation of tracked order vibration is called a vibration signature. A signature may be constructed, for example, from 1LP vibration amplitudes (i.e., the fundamental harmonic of vibration of the low pressure shaft) measured as a function of the speed of the LP shaft.

Figure 2.5 shows an example signature for the amplitude of the 1LP tracked order, in which the distribution of vibration amplitudes acquired during a single flight of an
The examplar engine is shown for each of $N_o = 400$ speed bins, covering the range of LP shaft speeds $\omega_{LP} = [0 \ 100] \%$ max. This thesis describes a probabilistic approach to the analysis of engine data, and so we will consider the distribution of vibration amplitudes throughout the speed range using probability density functions (pdfs). The distribution of vibration amplitudes $|x|$ stored for each speed bin has been normalised such that it integrates to unity, giving an approximation of $p(|x|)$ in each speed bin. In this example, it may be seen that 1LP amplitude generally increases with increasing shaft speed up to $\omega_{LP} < 90\%$, and generally decreases for shaft speeds $\omega_{LP} > 90\%$. Figure 2.6 illustrates this normalisation by showing histograms of the tracked order amplitudes contained in the two speed bins indicated by the dashed white lines in Figure 2.5. The lower of the two speed bins, $\omega_{LP} = 40\%$, contains 194 tracked order amplitudes, and the upper of the two speed bins, $\omega_{LP} = 88\%$, contains 13,928 tracked order amplitudes. Note the differences in the $y$-axis between Figure 2.6(a) and (b). After normalisation, the amplitudes of the two distributions are within an order of magnitude of one another, as shown in Figure 2.6(c), allowing both to be plotted on the same axes. This probabilistic approach effectively removes the dependency of the analysis on the absolute number of samples observed at each point of the speed range.

### 2.6 Application of Novelty Detection

#### 2.6.1 Monitoring of developmental engines

The development programmes for modern jet engines last many years, with several months spent testing prototype engines in ground-based testing facilities. Eventually, the new engines are flown on test aircraft for evaluation in flight, as described in Chapter 1. Developmental engines, both on the ground or in flight, are heavily instrumented, and there is no tight constraint on computing resources or communications bandwidth.

Prototype engines are over-instrumented in comparison to in-service engines. Thus, there is usually a large amount of data available for constructing models of normality for monitoring development engines.

The monitoring of developmental engines is taken to be a high-bandwidth problem for the remainder of this thesis.
Figure 2.5: A signature of the amplitude of the 1LP tracked order $|x|$ against LP shaft speed, $\omega_{LP}$, where the colour scale on the right-hands side of the plot indicates the value of the distribution function $p(|x|)$ in each speed bin. The signature in this example has $N_\omega = 400$ speed bins over the range of LP shaft speeds [0 100]% max.

Figure 2.6: (a), (b) Distribution of 1LP tracked order amplitudes for the two speed bins indicated by dashed white lines in Figure 2.5. (c) Corresponding pdfs, $p(|x|)$, for both bins after normalisation such that each integrates to unity.
2.6.2 Monitoring of in-service engines

The monitoring of engines used in either commercial or military service ("in-service engines") is severely constrained by limited communications bandwidth. Each EHM system has a communications pathway from it to ground-based stations, for the transmission of data allowing engine performance to be tracked remotely. If these communications pathways were high-bandwidth channels, it might be possible to transmit large quantities of data from the EHM system for ground-based analysis. While such high-bandwidth communications have been proposed for installation in EHM systems in the long-term (where industry experts have suggested a timescale of the next five-to-ten years), existing communications pathways have extremely limited bandwidth (ARINC 2005). Such transmissions are typically of the order of several kilobytes in existing communications systems, due to the expense incurred by transmitting data through the satellite networks that form much of the communications infrastructure.

The monitoring of in-service engines is taken to be a low-bandwidth problem for the remainder of this thesis.

2.7 Methodology

In both ground-based testing and in-flight use aboard a test-bed aircraft, the operational cycle comprises activation of the engine and operating at "idle" engine speed, followed by a period of use at higher engine speeds providing thrust, followed by a return to "idle" engine speed, at which the engine operates until it is either deactivated, or another operational cycle is initiated. Each such cycle is described as a “run” in this thesis, whether it be a ground-based testing cycle, or a flight aboard a test-bed aircraft.

In previous work, novelty detection has typically been performed using a model of normality that is constructed using data obtained from a large population of systems of a similar type to those that we wish to monitor. We term this approach population-generic novelty detection.

For example, Hann (2008) constructed a model of normality using vital-sign data from a large population of high-risk hospital patients, and then used it to perform novelty detection in test data from previously-unseen patients. Similarly, Nairac et al. (1999), Hayton et al. (2000), and Clifton et al. (2007) constructed models of normality
using jet engine vibration data from runs of a large population of gas-turbine engines, which were then used for (population-generic) novelty detection using test data from previously unseen engines.

This thesis considers *engine-specific* novelty detection, in which data from runs of the engine being monitored are used to construct an engine-specific model of normality, against which previously unseen data from that same engine are then tested.

**Motivation for an engine-specific approach**

Existing EHM systems (Hayton et al. 2003) take a population-generic approach, in which a single model is used to detect abnormal engine vibration for all engines of a particular model type. Every time a new engine model is designed, a new population-generic model needs to be constructed, and its parameters set such that it performs acceptably for all engines of that type.

However, there are significant differences in “normal” vibration characteristics between engines of the same type. Engine experts refer to engines as being “rough” or “smooth” depending on the magnitude of their vibration. An engine which is more rough than the engine for which data is shown in Figure 2.5, for example, would have vibration distributions shifted higher on the $y$-axis. Conversely, an engine which is more smooth than the exemplar engine would have vibration distributions shifted lower on the $y$-axis. In order to create an EHM system that has an acceptably low false alarm rate, alarm thresholds for a population-generic model must be set very high, in order to cope with particularly rough engines. This results in the EHM system only being able to detect engine events which correspond to very large amplitude of vibration.

By comparison, the engine-specific approach taken by the work described in this thesis learns the behaviour of individual engines, rather than being constrained to cope with the most rough engines, allowing alarm thresholds to be set much lower than those of population-generic systems. The use of lower thresholds allows for the detection of engine events which are much smaller in vibration amplitude than those of conventional systems. This is particularly important for the detection of subtle pre-cursor events to eventual engine failure.

This engine-specific approach requires extension of the training procedures conventionally used for the population-generic approach, which we discuss below.
2.7.1 Population-generic novelty detection

A novelty detection scheme employs a model of normality $M(\theta)$, where $\theta$ are the free parameters of the model. The model is used to assign novelty scores $z(x)$ to data $x$, where larger novelty scores $z(x)$ correspond to increased “abnormality” with respect to the model of normality. A novelty threshold $z(x) = \kappa$ is defined such that $x$ is classified “normal” if $z(x) \leq \kappa$, or “abnormal” otherwise. Different types of models $M$ used in novelty detection, methods for setting their free parameters $\theta$, and methods for determining novelty thresholds $\kappa$, are considered in Chapter 3.

Train-validate-test

The methodology conventionally used to perform population-generic novelty detection is a three-step procedure, in which the available data are partitioned into three independent datasets, as shown in Figure 2.7.

- **Train**: A training set of “normal” examples from the global population is used to determine the free parameters of a model of normality $\theta$, as described in Chapter 3.

- **Validate**: A validation set of “normal” and “abnormal” examples is used to help select the complexity of the model $M(\theta)$, and to set the novelty threshold $z(x) = \kappa$, as described below. Validation is typically performed in parallel with the training step.

- **Test**: A test set of “normal” and “abnormal” examples is subsequently used to evaluate independently the performance of the trained novelty detection system, as described below.
Figure 2.8: A confusion matrix tabulates the number of true and false negative and positive novelty detections, with reference to their required classification.

Measuring the performance of a novelty detection system

The validation and testing steps require some method of evaluating the capability of the novelty detection system to classify correctly “normal” and “abnormal” examples. This is typically performed using a confusion matrix, as shown in Figure 2.8.

The figure shows a confusion matrix that compares classifications made by the novelty detection system (on the vertical axis) to the actual, required classifications. If the novelty detection system classifies an example as “abnormal” with respect to the model of normality $M(\theta)$, this is a true-positive (TP) or false-positive (FP) classification depending on whether the example is actually “abnormal” or “normal”, respectively. Conversely, if the novelty detection system classifies an example as “normal”, this is a true-negative (TN) or false-negative (FN) classification depending on whether the example is actually “normal” or “abnormal”, respectively.

The figure shows four measures often used to characterise the classification accuracy of a novelty detection system, where measures are usually taken in pairs $(S_n, S_p)$ or $(PPV, NPV)$. Each pair contains opposing constraints: sensitivity $S_n$ can be increased at the expense of decreased specificity $S_p$, or vice versa, and PPV can be increased at the expense of decreased NPV, and vice versa.

---

3N.b., confusion matrices are applicable for evaluating the performance of any system that has a binary outcome in comparison to some desired binary outcome. A novelty detection system, with the binary outcome of “normal” and “abnormal”, is one such example.
However, for the case of novelty detection, we can relate the confusion matrix to the statistical framework of hypothesis testing. In novelty detection, our goal is to test the hypothesis $H$ that some example $x$ is “abnormal” with respect to a model of normality $M(\theta)$; i.e., that $x$ was not generated by $M(\theta)$. We define a null hypothesis $H_0$ that $x$ is “normal”; i.e., that $x$ was actually generated by the distribution $M(\theta)$. Two types of error can then be made:

- **Type I error**: the null hypothesis $H_0$ is rejected when it is actually true; i.e., $x$ was generated by $M(\theta)$, and is “normal”, but the novelty detection system classified it “abnormal”. This is also called an $\alpha$-error, or, in terms of the confusion matrix, a FP.

- **Type II error**: the null hypothesis $H_0$ was not rejected when it was actually false; i.e., $x$ was not generated by $M(\theta)$, and is “abnormal”, but the novelty detection system classified it “normal”. This is also called a $\beta$-error, or, in terms of the confusion matrix, a FN.

We can quantify the probability of the Type I and II errors occurring by defining

False Positive Rate, $\alpha = P($making a Type I error$)$
\[
\alpha = \frac{FP}{TN + FP} = 1 - Sp
\] (2.5)

\[
\text{False Negative Rate, } \beta = P($making a Type II error$)$
\[
\beta = \frac{FN}{TP + FN} = 1 - Sn
\] (2.6)

which, like the measures shown in Figure 2.8, form a pair of opposing constraints: $\alpha$ can be increased at the expense of decreased $\beta$, and vice versa$^4$.

Based on one of these measures, the validation and testing steps can be used to estimate the classification performance of the novelty detection system, which results in the advantages described in the next subsection.

---

$^4$In statistical literature, the “power” is defined to be $P_w = 1 - \beta$. From (2.6), $\beta = 1 - Sn$, thus $P_w = Sn$, and so power is equivalent to sensitivity.
Benefits of train-validate-test

The train-validate-test procedure has several advantages:

(i) The use of an independent validation set can be used to help select the complexity of model $M(\theta)$ during training; i.e., to set the number of degrees of freedom in $\theta$. Increasing the complexity of the model should result in a better fit of the model to the training data, because as more degrees of freedom are permitted, the distribution of “normal” data is learned more accurately. However, increasing the complexity too much will cause the model to learn only the training data themselves, rather than the characteristics of the underlying process that is assumed to have generated them. This is over-fitting\(^5\), which results in poor novelty detection performance when applied to previously unseen data. Thus, the complexity of the model must be constrained in some way to prevent over-fitting.

The use of a validation set that is independent of the training set provides an estimate of the model’s fit to previously unseen data during the training process. Model complexity can be increased up to the point at which the accuracy of the model’s fit to the independent validation set begins to decrease.

(ii) Similarly, an independent validation set provides an indication of which value $\kappa$ should take to perform novelty detection optimally with previously unseen data. If $\kappa$ were set using the training set, which in novelty detection is comprised only of “normal” examples, there would be no way of knowing how well the novelty detection system would perform when used in practice with previously unseen data.

(iii) The test set provides an objective measure of how the novelty detection system will perform in practice, using the $M(\theta)$ and $\kappa$ that have been determined during training and validation, and which remain fixed for the subsequent testing.

Chapter 6 describes techniques of cross-validation that may be used to provide the benefits described here even if there are insufficient data to populate the three independent sets (training, validation, and test sets).

\(^5\)Alternatively referred to as having a model that does not “generalise” successfully to previously unseen data.
2.7.2 Engine-specific novelty detection

Engine-specific novelty detection requires that a new model of normality is constructed for each engine, so that data from that engine are compared with a model derived from previously acquired data from the same engine, rather than with a population-generic model. We wish to retain the advantages of the train-validate-test procedure, described in the previous subsection, but must now extend it for use with engine-specific novelty detection.

Population-generic novelty detection uses training and validation sets comprising data from many engines, and thus the model $M(\theta)$ and the novelty threshold $\kappa$ may be determined in advance of deployment into an EHM system. In contrast to this, engine-specific monitoring requires that data are acquired from the engine being monitored, and then a model of normality $M(\theta_n)$ is constructed from those data. Here, we have added the subscript $n$ to denote that model parameters $\theta_n$ are specific to engine $n$. This is an “on-line” procedure, in that acquisition and model construction must occur during the operational life of the engine.

Before we describe the extension of the train-validate-test procedure to engine-specific novelty detection, we must first consider how “on-line” novelty detection would be performed.

On-line novelty detection

On-line novelty detection involves the construction of a model of normality $M(\theta_n)$ from engine data during the operational life of that engine. This could be achieved by performing the conventional train-validate-test procedure (illustrated in Figure 2.7) using data acquired from engine $n$ over a predetermined number of flights. However, engine vibration characteristics are known to change through time, such as the gradual (but “normal”) deterioration caused by engine wear. We wish to allow the model parameters to change during the service life of the engine to reflect these gradual changes in “normal” engine condition. Thus, we denote the model $M(\theta_{n,t})$ to indicate that the parameters $\theta$ can vary through time $t$. The approach taken in the work described in this thesis is illustrated in Figure 2.9.

After run $r$ of the engine, a model of normality is constructed using a training set
Figure 2.9: Training and testing using data acquired from the monitored engine. The training and test windows are shown underneath as black and grey lines, respectively. As more runs are performed, the training set contains data from increasingly large numbers of runs (shown in green in the figure). However, data from run 7 are classified “abnormal” with respect to the model of normality at that stage (shown in red in the figure), and so are not added to the training set. Note that the “normal” and “abnormal” classifications (shown by green and red in the figure) are those decided upon by the online novelty detection system.

of data acquired during runs 1...r. Novelty detection is performed using data acquired during the next run r + 1 using this model of normality. If the new data from run r + 1 are deemed “normal” with respect to the model of normality, they are included in the training set, and used to construct a new model of normality. In the figure, data acquired during runs r = 1...6 are classified “normal”, and so added to the training set as each run is completed. The model of normality is updated every time the training set changes. However, if the new data from run r + 1 are deemed “abnormal” with respect to the model of normality, they are not added to the training set, and the model of normality is not updated. In the figure, data acquired during run 7 are classified “abnormal”, and thus they are not added to the training set.

After an “abnormality” such as that shown in run r = 7, two outcomes could occur:

- The abnormality was a significant engine event, typically resulting in maintenance of the engine. Maintenance actions can significantly alter the operational characteristics of an engine, and so a new model of normality should be learned from subsequent flights, as if the engine were new. This is not a scenario considered in this thesis.
• The abnormality was temporary, due to a transient precursor event. Subsequent flights may return to normality, and so the existing model of normality is retained. This is the mode of operation investigated in a number of case studies in this thesis. The labelling of a run takes place automatically; the example shown in the figure contains a transient engine event during run $r = 7$, and so for run $r = 8$, the training set (and thus the model of normality) remains unchanged from run $r = 7$. In the example shown by the figure, data from runs $r > 7$ are classified as “normal”, and thus are all added to the training set.

**Extension of train-validate-test for engine-specific novelty detection**

In order for a novelty detection system to be used in an engine-specific manner, we must decide on a suitable value for the novelty threshold if a probabilistic novelty score is used; e.g., $\kappa = 0.99$ or $\kappa = 0.999$. While the model of normality $M(\theta_{n,t})$ will change for each engine because we employ engine-specific novelty detection, we assume that the probability with which we expect “abnormal” events to occur should be similar for all engines. Thus, we will select a single value of $\kappa$ for use with all engines. In order to determine this value, we extend the train-validate-test procedure. This requires engine-specific data, such as that illustrated in Figure 2.9, with the addition of expert labels for the data against which the performance of novelty detection can be compared. Figure 2.10 shows the approach taken in the work described in this thesis.

Available datasets from a number of engines are partitioned into (i) those to be used for training and validation, and (ii) those to be used for testing. The latter partition is further subdivided into those datasets that contain examples of “abnormal” engine behaviour, and those that contain only “normal” examples. In practice, the latter are available in greater quantity than the former, because examples of “abnormal” behaviour are rare in comparison to examples of “normal” behaviour for jet engines, as these safety-critical systems very rarely have abnormal events, even during the development stage.

Note that each dataset is acquired from a different engine (e.g., from 8 different engines in Figure 2.10: $A_1$, $A_2$, $B_1$, $B_2$, and $C_1$ to $C_4$), though engine-specific novelty detection is performed with the dataset specific to that engine, as illustrated in

---

6The selection of a single value of $\kappa$ for all engine-specific models corresponds to selecting the significance level (or “$p$” level) in conventional statistical testing.
Figure 2.10: Train-validate-test for engine-specific novelty detection. Available datasets are partitioned into (A) datasets used for training and validation, comprising “normal” and “abnormal” runs, coloured green and red, respectively; (B) datasets used for testing, containing some “abnormal” runs; (C) datasets used for testing, containing only “normal” runs. Here, for example, dataset $A_1$ comprises data from 57 runs, where runs 49...57 were deemed “abnormal”. Datasets containing examples of “abnormal” engine behaviour are divided equally between (A) training/validation and (B) testing. All datasets containing no examples of “abnormal” engine behaviour are placed in (C).

Figure 2.9. The following two steps are performed:

- **Training and validation**: expert labels are used with the training and validation datasets ($A_1$ and $A_2$ in the figure) to optimise $\kappa$. The value of $\kappa$ is determined such that the occurrence of $\alpha$ and $\beta$ errors (FP and FN errors) is minimised.

- **Testing**: expert labels are used with the test datasets ($B_1$, $B_2$ and $C_1$ to $C_4$ in the figure) to calculate $\alpha$ and $\beta$ statistics, using the value of $\kappa$ fixed during the training/validation step.

The test datasets containing examples of “abnormal” behaviour that have been labelled by engine experts (datasets $B_1$ and $B_2$ in Figure 2.10) are used to evaluate both the $\alpha$ and $\beta$ (FP and FN) error rates. The test datasets containing only
examples of “normal” behaviour (datasets \(C_1, \ldots, C_4\) in Figure 2.10) are used to evaluate only the \(\alpha\) error rate.

Note that, unlike population-generic novelty detection in which the model of normality \(M(\theta)\) is fixed along with \(\kappa\) after training and validation, in this engine-specific approach, on-line training is used for each dataset (i.e., Figure 2.9 applies equally to \(C_1, \ldots, C_4\) as to \(A_1, A_2, B_1,\) and \(B_2\)).

2.8 Methodology for High- and Low-Bandwidth Methods

2.8.1 High-bandwidth methodology

The high-bandwidth problem allows a sample-by-sample approach, whereby novelty detection takes place during the engine run. This approach has the advantage that “abnormal” engine behaviour can be detected during engine use, and a run can be interrupted if an example of “abnormal” behaviour is identified. This is of particular benefit in the monitoring of developmental engines, in which the prototype engines are often unique and require as much early warning of deterioration as can be provided.

Chapters 5 and 6 describe novel high-bandwidth methods for novelty detection in data obtained from developmental engines. Models of normality are constructed from vibration amplitudes of tracked orders and other features derived from vibration spectra, such as the total spectral power in each vibration spectrum. A multivariate approach is taken, in which models of normality characterise the distribution of many features simultaneously, rather than considering each independently. This has the advantage that changes in the subtle relationships that may exist between features can be identified, which is not possible when a univariate approach is taken.

2.8.2 Low-bandwidth methodology

In the case of the low-bandwidth approach for monitoring in-service engines, the full quantity of available data from each engine run must be summarised into a smaller representation that retains sufficient distinction between “normal” and “abnormal” engine behaviour such that novelty detection may still be performed. The vibration signature
introduced in Section 2.5 forms the basis for this representation, as will be described later in this thesis.

At the end of an engine run, a summary representation is constructed from the data observed throughout that run. It is then compared with the model of normality constructed from the training set comprising data from previous runs. If it is deemed to be “normal” with respect to that model, the training set is updated to include it, and a new model of normality is constructed, as shown in Figure 2.9.

Thus, novelty detection may take place using much smaller amounts of computer processing resources and memory storage than would be required for high-bandwidth approaches, and within tight bandwidth limitations. This occurs at the cost of only being able to perform novelty detection at the end of each run, rather than during each run, as is possible with high-bandwidth approaches.

Chapters 8 and 9 describe a novel low-bandwidth method for novelty detection with data obtained from in-service engines, such as is suitable for novelty detection in a practical EHM system, given the very tight constraints on communications bandwidth that were described in Section 2.6.2.
Chapter 3

Methods of Novelty Detection

3.1 Introduction

This chapter considers methods for constructing a model of normality \( M(\theta) \) from a training set of appropriately normalised \( Q \)-dimensional data \( \mathbf{X} = \{ \mathbf{x}_1, \ldots, \mathbf{x}_N \}, \mathbf{x} \in \mathbb{R}^Q \). Each of the \( Q \) dimensions of data \( \mathbf{x} \) takes the value of a feature (such as the value of a tracked order, as described in Section 2.4), and thus \( \mathbf{x} \) is alternatively called a feature vector. The \( Q \)-dimensional space \( \mathbb{R}^Q \) occupied by the feature vectors is called the data space, or occasionally the input space.

Novelty scores \( z(\mathbf{x}) \) are assigned to data \( \mathbf{x} \). We consider how a novelty threshold \( \kappa \) may be defined on \( z \) such that \( \mathbf{x} \) is classified “normal” if \( z(\mathbf{x}) \leq \kappa \) and “abnormal” otherwise. Thus, \( z(\mathbf{x}) = \kappa \) defines a decision boundary.

We first consider distance-based novelty detection, in which \( z(\mathbf{x}) \) is determined according to the distance (in some sense) of \( \mathbf{x} \) from the model of normality. We then consider principled probabilistic methods of novelty detection, in which \( z(\mathbf{x}) \) is determined according to the probability of \( \mathbf{x} \) being “normal” with respect to the model of normality.

3.2 Distance-Based Novelty Detection

Distance-based novelty detection techniques may be broadly partitioned into two types:

- those classical techniques in which distance from \( \mathbf{x} \) to the model of normality in \( Q \)-dimensional space is used to determine \( z(\mathbf{x}) \). These typically use the Euclidean or Mahalanobis distance;
• techniques in which the data are mapped into some higher \( R \)-dimensional space \((R > Q)\) using a transformation \( \Psi(x) \), and where the distance between \( \Psi(x) \) and the \( R \)-dimensional model of normality is used to determine \( z(x) \).

This section introduces the most popular methods from each type, and considers their respective advantages and disadvantages.

We note in passing that distance-based techniques are often referred to as boundary techniques in the literature (Luo 2006, Clifton 2007, Timusk et al. 2008, Guo et al. 2009), so called because the (distance-based) novelty threshold \( z(x) = \kappa \) defines a boundary around the model of normality, within which \( z(x) < \kappa \) and outside of which \( z(x) > \kappa \). However, this concept can be applied to all novelty detection techniques that define a novelty threshold based on a novelty score, because all result in a decision boundary surrounding the model of normality, and thus even probabilistic techniques with novelty boundary \( p(x) = \kappa \) could be considered to be “boundary” methods using this definition.

The term has also been more satisfactorily applied to refer specifically to methods in which \( z(x) \) is defined in terms of the boundary alone, such as Support Vector Machines (SVMs), which are considered in Section 3.2.2.

3.2.1 Classical distance-based methods

Perhaps the simplest and most intuitive of all novelty detection methods, classical distance-based approaches define \( z(x) \) in terms of the Euclidean distance from \( x \) to the model of normality defined by a set of “prototype” data \( \mu_k \), \( k = 1 \ldots K \). In the most elementary methods, \( \mu_k = X \), and \( K = N \); i.e., the training data are used as the prototypes. In less simplistic methods, \( K < N \), and \( \mu_k \) are typically determined using clustering methods.

Examples of classical distance-based methods include:

• the nearest neighbour (NN) algorithm (Duda et al. 2001), in which \( z(x) = \|x, \mu_k\|_2 \), and where \( k = \text{argmin}_k \|x, \mu_k\|_2 \). That is, \( z(x) \) is the Euclidean distance between \( x \) and the nearest \( \mu_k \).

• variants of the \( K \)-nearest neighbour algorithm for novelty detection, such as that proposed by Knorr & Ng (1998), in which \( z(x) \) is defined to be the number of the \( K \) nearest-neighbours \( \mu_k \) for which \( \|x, \mu_k\|_2 < \alpha \), for some constant distance \( \alpha \).
variants of the NN algorithm, in which $z(x) = \frac{1}{\sigma_k} \|x, \mu_k\|_2$ for some cluster width $\sigma_k$. In Nairac et al. (1997), $\sigma_k$ is defined to be the average of the distances between the $N_k$ data in cluster $k$ and that cluster centre $\mu_k$; i.e., $\sigma_k = \frac{1}{N_k} \sum_{j \in k} \|x_j, \mu_k\|_2$. In Clifton et al. (2006a) and Clifton et al. (2006b), $\sigma_k$ is defined to be the standard deviation of those intra-cluster distances; i.e., $\sigma_k = \left(\frac{1}{N_k} \sum_{j \in k} (x_j - \mu_k)^2\right)^{1/2}$.

Such methods are heuristic and typically require the selection of parameters using manual methods, removing the possibility of using them for automatic construction of a model of normality, though approaches have been suggested for parameter selection using semi-automated techniques (Tarassenko et al. 1999, Tarassenko et al. 2000).

Furthermore, because the resultant novelty boundary $z(x) = \kappa$ is based on a distance from the model of normality, the decision boundary is piecewise-hyperspherical\(^1\), which is unlikely to be an accurate representation of the true density of the training data.

### 3.2.2 Other distance-based methods

The SVM is a popular technique for forming decision boundaries that separate data from $J$ known classes $C_1, \ldots, C_J$, and was first proposed by Boser et al. (1992), and later developed in Schölkopf (1997), Burges (1998), Vapnik (1998), and Vapnik (2000), among many others.

In the original, most basic formulation of SVM, a decision boundary is found that best separates two classes of data in the $Q$-dimensional space of the training data. An objective function is selected defining the hyperplane that separates the data with maximum distance between the two classes (referred to as the maximum margin in SVM literature), as shown in Figure 3.1.

Techniques of quadratic programming (Cristianini & Shawe-Taylor 2000, Fletcher 1988) are used to solve what then becomes a convex optimisation problem in order to determine the decision boundary.

The resultant decision boundary implicitly defines a decision function $f(x) : \mathbb{R}^Q \rightarrow \mathbb{Z}$ that minimises the risk function

$$R[f] = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} |f(x_i) - y_i|$$

\(^1\)E.g., piecewise-circular, in the case of a 2-D model of normality.
Figure 3.1: Example SVM decision boundary (shown as a dashed line) that separates 2-dimensional data of class ‘◦’ and of class ‘×’ with maximum margin; i.e., the margin between the decision boundary and the convex hull of the two classes is maximised. (The convex hull of a set of points X is defined to be that minimal convex region of space that contains all points in X.)

where the true class of \( x_i \) is \( y_i : \{C_1, \ldots, C_J\} \). The SVM grew out of statistical learning theory (Vapnik 1998, Vapnik 2000), which is typically concerned with minimising misclassification risk. For the two-class problem for which \( y : \{-1, 1\} \), the equation of the decision boundary is that of the hyperplane:

\[
x_i \cdot w + b = 0
\]  
(3.2)

where \( w \) is normal to the hyperplane, and \(|b| / ||w||\) is the distance from the hyperplane to the origin. The hyperplane parameters \( w \) and \( b \) are given by

\[
w = \sum_{i=1}^{N_s} \alpha_i y_i s_i
\]
(3.3)

\[
b = \frac{1}{N_s} \sum_{j=1}^{N_s} \left( \frac{1}{y_j} - w \cdot s_j \right)
\]
(3.4)

\[
b = \frac{1}{N_s} \sum_{j=1}^{N_s} \left( \frac{1}{y_j} - \sum_{i=1}^{N_s} \alpha_i y_i s_i \cdot s_j \right)
\]
(3.5)

which depend only on a subset of the training vectors \( s_i \subseteq X \), termed the support vectors (of which there are \( N_s \)), with associated constants\(^2\) \( \alpha_i \). Thus the decision boundary is

\(^2\)Obeying the constraints \( \alpha_i \geq 0, \forall i \) and \( \sum_i \alpha_i y_i = 0 \)
defined only by those vectors from \( \mathbf{X} \) closest to it; i.e., the support vectors. The location of the boundary does not take into account the distribution of the majority of data that lie further from the boundary.

Initially, the case in which classes are linearly separable was considered (Vapnik & Lerner 1963, Vapnik & Chervonenkis 1964), as shown by the example in Figure 3.1. However, classes are seldom linearly separable, and so the above SVM formulation was expanded to the more general case in which classes are allowed to overlap in feature space to some limited degree\(^3\).

This allowed a greater number of problems to be tackled, using linear boundaries in the \( Q \)-dimensional feature space of the data. In practice, however, many problems involve classes that overlap to a significant degree, which cannot be linearly separated, even allowing for some limited degree of overlap.

The linear method of classification was therefore further generalised to a non-linear method (Boser et al. 1992), in which \( Q \)-dimensional data \( \mathbf{x} \) are mapped into a higher \( R \)-dimensional space \( (R > Q) \) using some kernel-based transformation \( \Psi(\mathbf{x}) \), with the goal of selecting \( \Psi(\mathbf{x}) \) such that the classes are linearly separable in the \( R \)-dimensional space, even if there is considerable overlap in the original \( Q \)-dimensional space. If we substitute for \( \mathbf{x} \) in (3.5), the equation for the hyperplane parameter \( b \) then becomes:

\[
b = \frac{1}{N_s} \sum_{j=1}^{N_s} \left( \frac{1}{y_j} - \sum_{i=1}^{N_s} \alpha_i y_i \Psi(s_i) \cdot \Psi(s_j) \right)
\]  

(3.6)

The above depends on a dot product \( \Psi(s_i) \cdot \Psi(s_j) \) in the high-dimensional space, and Mercer’s theorem (Courant & Hilbert 1959, Vapnik 1998) states that we can use some kernel \( K(s_i, s_j) = \Psi(s_i) \cdot \Psi(s_j) \) to write

\[
b = \frac{1}{N_s} \sum_{j=1}^{N_s} \left( \frac{1}{y_j} - \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, s_j) \right)
\]  

(3.7)

which is the so-called **kernel trick**: the nature of the \( \Psi(\mathbf{x}) \) transformation and the space into which it maps the data need not be considered, because we need only evaluate kernel \( K \). That is, the use of a kernel defines an implicit high-dimensional transformation, which may even be of infinite dimensionality. Typical choices of kernel

---

\(^3\)By allowing a certain proportion of training data to fall on the “wrong” side of the decision boundary, achieved using a “slack variable” in the method’s objective function.
$\mathcal{K}(\mathbf{x}_1, \mathbf{x}_2)$ include the polynomial, $(1 + \mathbf{x}_1^T \mathbf{x}_2)^\alpha$ for some polynomial order $\alpha$, the Gaussian, $\exp(-|\mathbf{x}_1 - \mathbf{x}_2|^2/\sigma^2)$ for some variance $\sigma^2$, and the sigmoid, $\tanh(\beta \mathbf{x}_1^T \mathbf{x}_2 - \delta)$ for some constants $\beta$ and $\delta$.

This non-linear formulation of the SVM has been applied to novelty detection in two related approaches. The first (Tax & Duin 1999) defines the novelty boundary as being a hypersphere in the high-dimensional transformed space that contains all of the training data (or some portion of it, allowing for a limited number of training data to fall on the “wrong” side of the boundary).

The second (Schölkopf et al. 1999) defines the novelty boundary to be that which separates the training data from the origin of the transformed space with maximum margin. Typically, a Gaussian kernel is used. This approach is motivated by the fact that $\mathcal{K}(\mathbf{x}_1, \mathbf{x}_2) \geq 0$ and $\mathcal{K}(\mathbf{x}_1, \mathbf{x}_1) = 1$ for all $\mathbf{x}_1$ and $\mathbf{x}_2$, and thus all transformed data $\Psi(\mathbf{x})$ lie in the positive region of transformed space. Hence, vectors similar to those in the training set take values of approximately unity in the transformed space, while dissimilar vectors tend towards the origin. Separating data that are dissimilar to the training data is achieved by separating the transformed training data from the origin.

It has been noted (Schölkopf et al. 1999, Clifton 2007) that the two SVM approaches to novelty detection provide equivalent results when the Gaussian kernel is used for transformation.

The SVM is used to determine scores $z(\mathbf{x})$ for test data $\mathbf{x}$ which are positive if $\mathbf{x}$ lies within the novelty boundary (and which are thus classified “normal”), or negative otherwise. Thus, the novelty boundary occurs at $z(\mathbf{x}) = 0$. As the purpose of the SVM is to separate classes, the novelty score has no typical use apart from consideration of its sign.

We note that the novelty score reaches $z(\mathbf{x}) = 1$ for test data lying on the plane parallel to the novelty boundary passing through the convex hull surrounding the training data\textsuperscript{4}. The support vectors $\mathbf{s}$ in a novelty detection SVM will have $z(\mathbf{s}) = 1$, and thus we observe that if the shortest distance from the support vectors to the novelty boundary (in transformed space) is $L$, the novelty score for data $\mathbf{x}$ lying distance $M$ from the novelty boundary is $z(\mathbf{x}) = M/L$; i.e., a novelty score $z(\mathbf{x}) = 2$ indicates that $\mathbf{x}$ lies

\textsuperscript{4}and, in a two-class SVM, reaches $z(\mathbf{x}) = -1$ for data lying on the equivalent plane for the opposite class.
twice as far from the novelty boundary as the support vectors. Thus, the novelty score is a distance-based score, but only in the transformed space to which we do not have access, because we use the kernel trick to avoid its consideration. This means that, in practice, the absolute value of $z(x)$ is not used as distances in the (potentially hidden) transformed space are uninformative, and hence only the sign of $z(x)$ is required for novelty detection.

As with two-class SVMs, the novelty detection SVMs determine the location of the novelty boundary using only those data that lie closest to it (in the transformed space); i.e., the support vectors. All other data from the training set (those that are not support vectors) are not considered when setting the novelty boundary. Hence, the distribution of data in the training set is not considered when determining the novelty boundary.

### 3.3 Probabilistic Novelty Detection

#### 3.3.1 Overview

An alternative approach to novelty detection is that in which the distribution of the training data is used to determine the location of the novelty boundary. The methodology adopted by SVMs and similar techniques may be defensible for multi-class classification, in which we are interested only in separating known classes and thus selecting a boundary between those classes is justifiable. Indeed, the optimal Bayesian decision boundary for separating multiple classes tends to lie in the areas of data space with lowest support for either class (Bishop 2006); i.e., where the probability density $p(x)$ is low, which typically occurs in the areas of data space that lie between classes.

In order to determine the location of the novelty boundary, we assume that the training data are generated from some underlying probability distribution $D$, and then attempt to estimate $D$ using the training data. Our estimate $\hat{D}$ is thus a model of normality. We may then set a novelty threshold using $\hat{D}$ in some manner, such that it has a probabilistic interpretation.

We note that the existence of a fixed underlying distribution is a strong assumption.

---

5 The proportion of the training data that may be selected by the training procedure to be support vectors is controlled by a single SVM parameter, the value of which is typically selected using empirical methods.
which we term *static novelty detection*. Fully *dynamic* novelty detection takes into account changes in the underlying distribution that occur through time, which is likely to be the case for most actual complex systems. In such cases, system dynamics change in response to gradually changing condition; for example, with component wear over time in mechanical systems, or homeostatic changes in biological systems.

### 3.3.2 Density estimation

Estimation of some underlying data density $D$ from multivariate training data assumed to be generated from it is a well-established field (Silverman 1986, Scott 1992). Broadly, these techniques fall into *parametric* and *non-parametric* approaches, in which the former assume that the data are generated from a distribution of known form, while the latter make no such assumption. For the purposes of novelty detection from a system about which we have little prior knowledge, it is appropriate to select a modelling approach that is as general as possible. This allows us to model arbitrary data distributions accurately, and that has previously motivated the choice of non-parametric approaches (Markou & Singh 2003a, Markou & Singh 2003b).

Opinion in the literature is divided as to whether various techniques should be classified as either parametric or non-parametric. For the purposes of providing a probabilistic estimate $\hat{D}$, Gaussian mixture models (GMMs) and kernel density estimators have proven popular. The former are typically classified as a parametric technique (Markou & Singh 2003a, Hann 2008), because of the assumption that the data are generated from a weighted mixture of Gaussian distributions, while the latter are typically classified as a non-parametric technique (Duda et al. 2001, Webb 2002), as kernel density estimators are closely related to histogram methods, one of the earliest forms of non-parametric density estimation. We will here briefly consider both techniques, and show that they are closely related, before going on to adopt the GMM for further use.

**Kernel density estimation**

Assuming that our data are distributed according to $\mathbf{x} \sim D = p(\mathbf{x})$, the probability that a sample $\mathbf{x}$ is drawn from some region $\mathcal{R}$ in data space is

\[
P(\mathbf{x}|\mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}) \, d\mathbf{x}
\]  

(3.8)
If we have a training set of $N$ samples, and each sample has a probability of $P(x|\mathcal{R})$ of lying in region $\mathcal{R}$, the binomial theorem can be used to determine the probability of the number of samples $N_{\mathcal{R}}$ that fall in $\mathcal{R}$:

$$
N_{\mathcal{R}} = \frac{N!}{N_{\mathcal{R}}!(N-N_{\mathcal{R}})!} P(x|\mathcal{R})^{N_{\mathcal{R}}} \{1 - P(x|\mathcal{R})\}^{1-N_{\mathcal{R}}}
$$

(3.9)

The mean and variance of the fraction of samples falling in $\mathcal{R}$ is (Bishop 2006):

$$
E[N_{\mathcal{R}}/N] = P(x|\mathcal{R})
$$

(3.10)

$$
\text{Var}[N_{\mathcal{R}}/N] = \frac{P(x|\mathcal{R})}{N} \times \{1 - P(x|\mathcal{R})\}
$$

(3.11)

where, for large $N$, this variance will be small, and so the distribution will be peaked around $E[N_{\mathcal{R}}/N]$, so:

$$
N_{\mathcal{R}} \approx N \times P(x|\mathcal{R})
$$

(3.12)

If $\mathcal{R}$ is small, then we can express the probability of $x$ falling in the volume $V_{\mathcal{R}}$ corresponding to region $\mathcal{R}$ in terms of the probability density $p(x)$ that we wish to estimate:

$$
P(x|\mathcal{R}) \approx p(x)V_{\mathcal{R}}
$$

(3.13)

and so, combining (3.12) and (3.13), we can write our density estimate:

$$
p(x) = \frac{N_{\mathcal{R}}}{N \times V_{\mathcal{R}}}
$$

(3.14)

If we fix $N_{\mathcal{R}}$ and determine $V_{\mathcal{R}}$ from our training data, we derive the $K$-nearest neighbour density estimator (Webb 2002), while if we fix $V_{\mathcal{R}}$ and determine $N_{\mathcal{R}}$ from the data, we derive the Parzen window estimator (Parzen 1962). Discounting the former due to its typically “noisy” estimation of $p(x)$ (Bishop 2006, Webb 2002), we further consider the latter.

We could fix volume $V_{\mathcal{R}}$ by defining a $Q$-dimensional hypercube of length $h$ around some centre $x$, where the membership function for some sample $x_i$, $i = 1 \ldots N$ would be defined to be

$$
\mathcal{K} \left( \frac{x - x_i}{h} \right) = \begin{cases} 
1 & \text{if } (x - x_i) \leq h/2, \\
0 & \text{otherwise}
\end{cases}
$$

(3.15)

The total number of samples falling within this hypercube centred at $x$ is then

$$
N_{\mathcal{R}} = \sum_{i=1}^{N} \mathcal{K} \left( \frac{x - x_i}{h} \right)
$$

(3.16)
which we can then substitute into (3.14) to give an estimate of the density at $x$

$$p(x) = \frac{1}{Nh^Q} \sum_{i=1}^{N} K \left( \frac{x - x_i}{h} \right)$$

where the volume $V_K = h^Q$. Rather than having one hypercube centred at $x$, we can also interpret (3.17) as representing the sum of a set of hypercubes, each centred at a different $x_i$.

$K(x, x_i)$ is an example of a kernel function, which must integrate to volume $V_K$ (i.e., such that it is unity after normalisation by its volume) and need not be a hypercube. We can think of the hypercube as being a special case of membership function, in which membership is strictly delimited by its finite boundaries. In fact, the membership of $K$ can be spread out arbitrarily, and a common choice of kernel is the Gaussian function, in which membership is spread out infinitely across data space, peaking at the centre of the kernel. Using the Gaussian kernel gives:

$$p(x) = \frac{1}{N(2\pi\sigma^2)^{Q/2}} \sum_{i=1}^{N} \exp \left\{ -\frac{||x - x_i||^2}{2\sigma^2} \right\}$$

Thus, we have a density estimate $p(x)$ formed from a mixture of Gaussian distributions with equal variance $\sigma$, each centred on one of the training data $x_i$, $i = 1 \ldots N$, which is termed the Parzen window estimate. Use of the Gaussian kernel has the advantage of providing a smooth resultant density estimate $\hat{D}$, rather than the discontinuous multivariate histogram provided by using the hypercube kernel, and this is more likely to represent the actual underlying distribution $D$. We note that as $\sigma \to 0$, the Parzen window estimate tends to the hypercube estimate with $h \to 0$.

Noting that all distributions within the mixture share the same variance $\sigma$, we can deduce that the smoothness of $\hat{D}$ can be influenced by varying that one parameter. This has the advantage that the model is simple to construct and control, but the disadvantage that some areas of data space may be over- or under-fitted by the model, as it is unlikely in practice that $D$ varies similarly in all areas of data space. Tarassenko et al. (1995) proposed a method of avoiding this, by allowing $\sigma$ to vary between limited partitions of data space, effectively by constructing separate Parzen windows estimates for each partition. This notion is akin to more recent mixture-of-experts methods in which separate models may be trained using data from differing regions of data space.
(Jacobs et al. 1991, Kittler et al. 1998, Avnimelech & Intrator 1999, Foggia et al. 1999), and then combined.

**Gaussian mixture models**

If we approach the density estimation problem from a parametric viewpoint, we arrive at mixture models which assume that $D$ can be approximated by a linear superposition of basic distributions (McLachlan & Basford 1988, McLachlan & Peel 2000):

$$p(x) = \sum_{k=1}^{K} P(k) \ p(x|\theta_k) \quad (3.19)$$

where $P(k)$ are the weights associated with each of the $p(x|\theta_k)$ components\(^6\), which are also the prior probabilities for each component. In order for (3.19) to define a probability distribution, it must integrate to unity, and so we constrain the priors

$$\sum_{k=1}^{K} P(k) = 1 \quad (3.20)$$

$$0 \leq P(k) \leq 1 \quad (3.21)$$

and ensure that the components are normalised probability distributions

$$\int p(x|\theta_k)dx = 1 \quad (3.22)$$

An appealing property of the mixture model to which we will return is that we can use Bayes’ theorem to determine the probability that data $x$ belongs to each of the $k = 1 \ldots K$ components

$$P(k|x) = \frac{p(x|k) \ P(k)}{p(x)} \quad (3.23)$$

where $P(k|x)$ is also termed the *posterior component probability*, or *responsibility*, which obeys

$$\sum_{k=1}^{K} P(k|x) = 1 \quad (3.24)$$

$$0 \leq P(k|x) \leq 1 \quad (3.25)$$

Typical choices of distribution for the components include the Gaussian (Nabney 2002), the gamma (Mayrose et al. 2005, Guillemot et al. 2005), the Student’s t (Svensen & Bishop 2005), the Poisson (Carvalho & Tanner 2007), and the Weibull (Heckman 6 also termed the *component likelihoods*.}
The Gaussian distribution is often used in the absence of prior information regarding the form of \( D \), and we will consider it further for its analytical properties when determining the location of a novelty threshold (see Section 3.4). Using the \( Q \)-dimensional Gaussian distribution, (3.19) becomes

\[
p(x) = \frac{1}{(2\pi)^{Q/2} \sum_{k=1}^{K} P(k)} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right\}
\]

where each of the \( k = 1 \ldots K \) Gaussian components is parameterised \( \mathcal{N}(\mu_k, \Sigma_k) \).

The equivalence of GMM and Parzen windows

We have previously observed that the Parzen windows estimator may be interpreted as a mixture of Gaussian distributions of equal variance \( \sigma \), each centred on one of the training data \( x_i \), \( i = 1 \ldots N \). If we write the equation of the GMM constraining the covariance matrices to be equal and isotropic \( \Sigma_k = \sigma^2 \), \( \forall k \), we have

\[
p(x) = \frac{1}{(2\pi \sigma^2)^{Q/2}} \sum_{k=1}^{K} P(k) \exp \left\{ -\frac{\|x - \mu_k\|^2}{2 \sigma^2} \right\}
\]

which, by comparison with the equation for the Parzen window estimator (3.18), we see that \( P(k) = 1/N \) and \( \mu_k = x_i \). That is, the Parzen window estimator is a special case of the GMM, given the following constraints:

- there are an equal number of Gaussian components as training data \( (K = N) \), with one component centred on each of the training data;
- each Gaussian component is isotropic with variance \( \sigma^2 \); and,
- each Gaussian component is weighted equally in the mixture.

We have previously seen how allowing \( \sigma^2 \) to vary by region of input space results in the Parzen window approach described by Tarassenko et al. (1995). When weights are allowed to vary between the Gaussian components, we have the weighted Parzen windows approach, proposed by Babich & Camps (1996). Vincent & Bengio (2003) proposed the use of general covariance matrices for individual components set according to neighbourhoods local to each kernel.

Nabney (2002) refers to both GMMs and related kernel estimators as “semi-parametric”, which agrees with the functional definition (Härdle et al. 2004), which states for any model that is a collection of distributions \( \{ P_\theta : \theta \in \Theta \} \) each with parameters \( \theta \),
• The model is parametric if its parameters can only take values from some finite subset of $\mathbb{R}^Q$; i.e., $\Theta \subset \mathbb{R}^n$.

• The model is non-parametric if its parameters can take values from some potentially infinitely-dimensional space $\mathbb{F}$; i.e., $\Theta \subset \mathbb{F}$.

• The model is semi-parametric if its has some parameters that are parametric, and others that are non-parametric; i.e., $\Theta \subset \mathbb{R}^Q \times \mathbb{F}$.

Thus, the GMM is semi-parametric because it has parametric elements $(\theta, \Sigma)_k$ and $P(k)$, and nonparametric elements because an arbitrary (potentially infinite) number of Gaussian components can be added to the model. While Parzen window estimators have been previously introduced using the notion that a Gaussian component is centred on each of the training data, this would rapidly lead to computationally intractable density estimation for large training sets. In practice (Hann 2008, Tarassenko et al. 2009), the number of components in a Parzen window model is limited by pre-clustering the original training set (of size $N$) to some representative set of $K$ prototypes ($K < N$), as described in Section 3.2.1.

In contrast to this use of pre-clustering followed by use of a Parzen window estimator, which is essentially a constrained GMM as we have seen, we opt to use a general GMM trained upon the full training set, the parameters of which are typically determined using expectation-maximisation (see below). The GMM is also a universal approximator in that it can model any density $p(x)$ to an arbitrary degree of accuracy using a finite number of Gaussian components (Park & Sandberg 1991).

3.3.3 Expectation-maximisation

In (3.19) we saw that the data likelihood with respect to a model with parameters $\theta$ is $p(x|\theta)$. Assuming that the data are i.i.d. (independent and identically distributed), the likelihood of the entire data set $X = \{x_1 \ldots x_N\}$ is defined to be:

$$L(\theta) = p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

(3.28)

where $L(\theta)$ is called the likelihood of the parameters given the data, or just the likelihood. We aim to find those parameters $\theta^*$ that maximise the likelihood; i.e., those
parameters that are most likely to have been responsible for generating the data that we have observed. This is the maximum likelihood method of parameter estimation, where typically the log-likelihood \( \ln L(\theta) = \ln p(X|\theta) \) is equivalently maximised for analytical convenience (Csiszár & Tusnády 1984, Hathaway 1986):

\[
\theta^* = \arg\max_\theta L(\theta)
\] (3.29)

If we knew which of the components \( C_k, k = 1 \ldots K \) generated each of the data \( x_i \), it would be straightforward to find a closed-form solution for the prior \( P(k) \), mean \( \mu_k \), and covariance \( \Sigma_k \) for each of the components, using those \( N_k \) data generated by each component \( x_i \in C_k \):

\[
P(k) = \frac{N_k}{N}
\] (3.30)

\[
\mu_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i
\] (3.31)

\[
\Sigma_k = \frac{1}{N_k} \sum_{x_i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T
\] (3.32)

We define labels \( Y = y_1 \ldots y_N \) to denote which of the \( K \) components generated data \( X = x_1 \ldots x_N \). These labels are not known and so we cannot find the parameters \( \theta^* \) directly. Rather than optimising the likelihood given the observed data \( L(\theta) \), we can instead optimise the likelihood given both the observed data \( X \) and the labels \( Y \). Not knowing the values of the labels \( Y \) explicitly, we can (i) estimate distributions describing their likely values according to the information available (the training data \( X \) and a current estimate of the model parameters \( \theta_{\text{old}} \)); i.e., we will estimate the distributions \( p(Y|X, \theta_{\text{old}}) \). Then, we can (ii) use those distributions over \( Y \) to find some new parameter estimate \( \theta_{\text{new}} \) that maximises our new likelihood \( L(\theta) \). Iterating over these two steps provides estimates of \( \theta \) that converge towards \( \theta^* \). This is the expectation-maximisation method (Dempster et al. 1977, McLachlan & Krishnan 1997), where steps (i) and (ii) are termed the e-step and m-step, respectively, for reasons explained later.
Formulating the EM method

Appendix A describes the formulation of the EM method. Step (i) is performed using equation (A.7):

\[
F(\theta, \theta_{\text{old}}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \left[ \ln p(x_i | \theta_k) + \ln P(k) \right] P(k | x_i) \tag{3.33}
\]

and step (ii) is performed using update equations (A.10), (A.11), and (A.12):

\[
P_{\text{new}}(k) = \frac{1}{N} \sum_{i=1}^{N} P(k | x_i) \tag{3.34}
\]

\[
\mu_{\text{new}} = \frac{\sum_{i=1}^{N} P(k | x_i) x_i}{\sum_{i=1}^{N} P(k | x_i)} \tag{3.35}
\]

\[
\Sigma_{\text{new}} = \frac{\sum_{i=1}^{N} P(k | x_i) (x_i - \mu_{\text{new}})(x_i - \mu_{\text{new}})^T}{\sum_{i=1}^{N} P(k | x_i)} \tag{3.36}
\]

Use of these update equations guarantees (by Jensen’s inequality) that the likelihood \( L(\theta) \) increases with each iteration (Bishop 2006), where typically the iterative procedure is terminated once the increase is below some threshold. We initialise the algorithm by using \( K \)-means clustering with \( K \) cluster centres to determine the initial labels \( Y \), and then using (3.30), (3.31), and (3.32) to determine \( \theta_0 \).

Further enhancements to the EM algorithm have been proposed such as the “greedy EM” algorithm, which uses the EM process to add new components to the mixture as required (Verbeek et al. 2003), while Bayesian (Roberts et al. 1998) and variational Bayesian techniques (Corduneanu & Bishop 2001, Beal & Ghahramani 2003) have been proposed in place of EM for learning model parameters.

### 3.4 Setting a Probabilistic Novelty Threshold

From a training set (obtained as described in Chapter 2), we construct a GMM-based model of normality using expectation-maximisation. We now need to determine a suitable novelty threshold. Within such a probabilistic approach, novelty scores may be defined as probability densities\(^7\) \( z(x) = p(x) \), and a typical approach to setting a novelty threshold is to threshold this value; i.e., \( p(x) = \kappa \). This method has been used for novelty detection in Barnett & Lewis (1994), Roberts & Tarassenko (1994), Tarassenko

\(^7\)Sometimes expressed as \(-\ln p(x)\) such that it increases with increasing novelty (i.e., decreasing probability density).
et al. (1995), Desforges et al. (1977), Tarassenko et al. (1999), Tarassenko et al. (2000), Clifton (2007), and Hann (2008), among others. Some authors have expressed GMMs as radial-basis function neural networks (Bishop 1994, Tarassenko 1998), and likened the thresholding of the output \( p(x) \) to that of thresholding the output of a neural network (Tax & Duin 1998).

The threshold on \( p(x) \) has no direct probabilistic interpretation: \( p(x) \) is simply a novelty score, and the threshold set such that separation between “normal” and any available “abnormal” data is maximised. These “abnormal” data are obtained from a validation set, as described in Chapter 2.

Some authors (Tarassenko et al. 1995, Nairac et al. 1997, Lauer 2001, Hann 2008) have interpreted the model output \( p(x) \) probabilistically, by considering the cumulative probability \( P \) associated with \( p(x) \); i.e., determining the probability mass obtained by integrating \( p(x) \) over the region \( R \) where the value of \( p(x) \) is above the novelty threshold \( \kappa \); i.e., the region \( \{ x \in R \mid p(x) \geq \kappa \} \):

\[
P(\kappa) = \int_R p(x) \, dx \tag{3.37}
\]

For a unimodal distribution, this equates to integrating from the mode of the probability density \( m = \arg\max_x p(x) \) to the probability contour defined by the novelty threshold \( p(x) = \kappa \), which can be determined in closed form for most regular distributions (e.g., the multivariate Gaussian distribution).

However, for the multi-modal \( p(x) \) which we will model with a mixture of Gaussian components, this may need to be performed using Monte Carlo techniques, as suggested in Nairac et al. (1997). An approximation in closed form for this was proposed by Larsen (2003).

Figure 3.2 shows a univariate, multimodal example, where \( p(x) \) has been estimated using two equivariant Gaussian kernels (variance \( \sigma_1^2 = \sigma_2^2 = 1 \)), with priors \( P(\mu_1) = 0.75 \), \( P(\mu_2) = 0.25 \). A novelty threshold is shown at \( p(x) = 0.05 \) in the figure, where the probability mass \( P_{0.05} \) enclosed by that threshold is shaded. This probability mass will fall in the range \( 0 \leq P_{0.05} \leq 1 \). In this example, the shaded areas from each Gaussian kernel do not overlap, and it is possible to perform the integration from (3.37) in closed form, treating each kernel independently. However, in the general case in which the
Figure 3.2: Integrating a bimodal probability distribution $p(x)$ to the contour $p(x) = 0.05$. The contour is shown as a horizontal dashed line, intercepting the distribution function at the locations shown by the vertical red lines. The area corresponding to the integration is shown in grey.

components of the mixture model overlap to a significant degree$^8$, the integration must be performed numerically.

Setting the novelty threshold in this way, and then determining the probability mass $P_\kappa$ enclosed by that $p(x) = \kappa$ contour, allows a probabilistic interpretation: if we were to draw one sample from the distribution $D = p(x)$, we would expect it to lie outside the novelty threshold $p(x) < \kappa$ with probability $1 - P_\kappa$. Thus, we could set the novelty threshold $p(x) = \kappa$ such that $P_\kappa$ is some desired probabilistic threshold; e.g., $P_\kappa = 0.99$.

**High-density regions**

Setting novelty thresholds at some value of $p(x) = \kappa$ as described above is closely related to the branch of functional theory dealing with *high-density regions* (HDR), proposed by Hyndman (1996)$^9$, in which we are interested in identifying regions of the domain of some function $f(\cdot)$ for which $f(\cdot) > 0$, also termed the “support” or “closure” of $f$.

---

$^8$In particular with a Parzen windows estimator, in which the components can overlap to a very great degree, such as when $\sigma$ is selected according using the method described by Bishop (1994) and Tarassenko et al. (2009).

$^9$Closely related to *high posterior density regions* (Box & Tiao 1973) and utilising the sorting of functions according to their probability density $p(x)$, which is termed *majorisation* theory (Marshall & Olkin 1983, Boshnakov 2003)
Indeed, Bishop (1994) refers to the task of novelty detection as that of determining the support of \( p(x) \), and Schölkopf et al. (2001) motivate their SVM approach to novelty detection as being that of determining high probability regions for finding the support of \( p(x) \).

In his original work, Hyndman (1996) uses the concept of the integration illustrated in Figure 3.2 to motivate a special form of box-plot (Tukey 1977) based on the probability mass enclosed by some \( p(x) = \kappa \) contour. In the example shown in the figure, this would correspond to placing two box-plots on the \( x \)-axis describing the distribution of data in each of the shaded regions, whereas a conventional box-plot would consist of a single box on the \( x \)-axis, with its centre placed at the centroid of \( p(x) \), and with inter-quartile ranges marked according to the univariate distribution over \( x \). The statistics of the HDRs are determined using the numerical integration techniques later described in Nairac et al. (1997).

HDRs have been used for novelty detection by setting thresholds using probability mass \( P \) in Paalanen et al. (2006) and Ilonen et al. (2006). This was combined with a SVM approach in Muñoz & Moguerza (1996), using the SVM novelty detections methods of Schölkopf et al. (1999).

3.5 Conclusions

We have shown the functional equivalence between Parzen window estimation and Gaussian mixture modelling for estimating the distribution of (“normal”) training data. We argue that GMMs are better suited to the problem of novelty detection because of their ability to be trained using a large training set, without recourse to data-set reduction techniques as are required for Parzen window estimation (such as \( K \)-means clustering).

The setting of novelty thresholds using the probability density \( p(x) \) has been considered, but we have shown that such methods do not allow a probabilistic interpretation. We have described how to set a novelty threshold using the cumulative density \( P \), such that a probabilistic interpretation may be possible.

Setting a novelty threshold such that \( P \) is equal to some probability mass, as described in Section 3.4, has associated disadvantages for novelty detection. In order to examine these disadvantages, we must first consider a different method for determining
the location of novelty thresholds: that of Extreme Value Theory.
Chapter 4
Extreme Value Theory

4.1 Classical Extreme Value Theory

EVT is a branch of statistics concerned with modelling the distribution of very large or very small values (extrema) with respect to a generative data distribution.

One of the first uses of EVT was in the field of civil engineering, for modelling the likelihood of observing extreme loads in structures with respect to a distribution of observed “normal” loads (Coles 2001). Here, we consider “classical” EVT as previously used in novelty detection (Roberts 1999, Roberts 2000, Worden et al. 2003, Sohn et al. 2005), in contrast to an alternative method commonly used in financial applications, often termed the peaks-over-threshold (POT) technique (Medova & Kriacou 2001). This latter method has proven particularly successful in the estimation of extreme events in financial data, such as the evaluation of credit risk, where it is used to determine the likelihood of large capital risks over some threshold. However, we will not consider its use for novelty detection in complex systems due to its requirement for sometimes-heuristic selection of many model parameters, often based upon prior knowledge of extreme values (Embrechts 2000).

4.1.1 Extreme value distributions

Consider a set of $m$ i.i.d. data, which we here assume to be univariate for the purposes of simplifying the discussion, $X = \{x_1, x_2, \ldots, x_m\}$, distributed according to some function $D(x)$, with maximum $x_{\text{max}} = \max(X)$. We define the distribution for $x_{\text{max}}$ to be $H(x_{\text{max}} \leq x)$; i.e., our belief in where the maximum of $m$ data drawn from distribution $D$ will lie is modelled by $H$.

According to the Fisher-Tippett theorem (Fisher & Tippett 1928) upon which clas-
classical EVT is based, $H$ must belong to one of the following three families of distributions:

**Type I (Gumbel)**

$$H_1(y) = \exp(-\exp(-y))$$  \hspace{1cm} (4.1)

**Type II (Fréchet)**

$$H_2(y) = \begin{cases} 
0 & \text{if } y \leq 0 \\
\exp(-y^{-\alpha}) & \text{if } y > 0 
\end{cases}$$ \hspace{1cm} (4.2)

**Type III (Weibull)**

$$H_3(y) = \begin{cases} 
\exp(-(-y)^\alpha) & \text{if } y \leq 0 \\
1 & \text{if } y > 0 
\end{cases}$$ \hspace{1cm} (4.3)

for $\alpha \in \mathbb{R}^+$, and where $y$ is a transformation of $x$, termed the *reduced variate*,

$$y = \frac{x - c}{d}$$ \hspace{1cm} (4.4)

for location and scale parameters $c$ and $d$, respectively. In classical EVT, these are dependent only on the number of data $m$ drawn from the underlying distribution $D$ (Embrechts et al. 2008):

$$c = \sqrt{2 \ln m} - \frac{\ln \ln m + \ln 4\pi}{2\sqrt{2 \ln m}}$$ \hspace{1cm} (4.5)

$$d = \frac{1}{\sqrt{2 \ln m}}$$ \hspace{1cm} (4.6)

Distributions (4.1), (4.2), and (4.3) are special cases of the Generalised Extreme Value (GEV) distribution (Castillo et al. 2005),

$$H_G = \begin{cases} 
\exp\left(- (1 + \xi y)^{-1/\xi}\right) & \text{if } \xi \neq 0 \\
\exp\left(- \exp(-y)\right) & \text{if } \xi = 0 
\end{cases}$$ \hspace{1cm} (4.7)

where $\xi \in \mathbb{R}$ is a shape parameter. If $\xi = (-\alpha)^{-1} < 0$ in the above, for some constant $\alpha$, we obtain the Weibull distribution (4.3). If $\xi = \alpha^{-1} > 0$ we obtain the Fréchet distribution (4.2), and if $\xi = 0$, we obtain the Gumbel distribution (4.1).

We here consider data distributed according to the single-sided Gaussian distribution, $X \sim |N(0,1)|$, for which the limiting EVD is the Gumbel distribution. We may also write this as the cumulative distribution function (cdf) describing our belief in the location of $x_{\max}$, conditional on the data $X$:

$$P(x_{\max} \leq x|X) = H_1(y) = \exp(-\exp(-y))$$ \hspace{1cm} (4.8)
which we will term \( P_e(x|X) \) for ease of notation. We obtain the corresponding pdf by differentiation, which we here call the *extreme value distribution* (EVD):

\[
p(x_{\text{max}} = x|X) = d^{-1}\exp\left(-y - \exp(-y)\right)
\]

which we will term \( p_e(x|X) \).

### 4.2 Setting Novelty Thresholds Using \( P \)

In Section 3.4, we described how to set novelty thresholds on some probability density \( D = p(x) = \kappa \) such that the probability mass “enclosed” by integrating within \( p(x) = \kappa \), using (3.37), was equal to some desired probability mass, \( P_\kappa \). For example, we could set a novelty threshold at \( p(x) = \kappa \) such that probability mass \( P_\kappa = 0.99 \) is enclosed. If a single sample \( x \) is then drawn from \( D \), its probability density will be \( p(x) \geq \kappa \) with a probability of \( P = P_\kappa = 0.99 \) (and it will thus be classified “normal”); the probability density will be \( p(x) < \kappa \) with a probability of \( P = 1 - P_\kappa = 0.01 \) (and thus the sample will be classified “abnormal”).

Supposing that \( D = N(0, 1) \), Figure 4.1 shows \( D \) (the black curve) with a novelty threshold set at \( P = 0.99 \) using the method described in Section 3.4. In the exemplar shown in the figure, setting the novelty threshold at \( P = 0.99 \) corresponds to setting the novelty threshold in data space at \( x_\kappa = 2.326 \). A single sample drawn from \( D \) will fall to the left of the novelty threshold (shown by the vertical dashed line) with probability \( P = 0.99 \), or will fall alternatively to the right of the novelty threshold with probability \( P = 0.01 \).

Using (4.9), we can determine the EVDs \( p_e(x) \) corresponding to drawing \( m = 10, 100, \) and 1000 samples from \( D \). Each EVD describes where we should expect the maximum of \( m \) samples drawn from \( D \) to lie on the \( x \)-axis. Note that as more samples are drawn, the expected location of their maximum increases, as shown by the EVDs which move higher on the \( x \)-axis for increasing \( m \).

Though the novelty threshold was set using the method described in Section 3.4, such that we would anticipate it to be exceeded with probability \( 1 - 0.99 = 0.01 \) when drawing a *single sample* (i.e., \( m = 1 \)), the probability that the novelty threshold is exceeded increases the more samples \( m \) we observe. For \( m > 1 \), the probability of exceeding the
Figure 4.1: The standard Gaussian distribution $x \sim D = N(0, 1)$, shown in black, with a novelty threshold set at $P = 0.99$ (shown as a dashed line). EVDs corresponding to $m = 10, 100,$ and $1000$ are shown in green, blue, and red, respectively, each describing where the maximum of $m$ samples drawn from $D$ will lie. As $m$ is increased, the novelty threshold set on $P$ is exceeded with increasing probability.

Figure 4.2: The probability of exceeding the exemplar novelty threshold set at $P = 0.99$, shown in black, as the number of samples $m$ drawn from distribution $D$ increases. Probabilities for $m = 10, 100,$ and $248$ are shown in green, blue, and red, respectively. At $m = 248$, the novelty threshold is exceeded with probability $1 - P_e(x) = 0.99$. 

51
novelty threshold is \( P(x > x_n) = 1 - P_e(x_n) \), where \( P_e(x_n) \) is the cumulative EVD for \( m \) samples evaluated at \( x = x_n \), determined using (4.8). That is, the probability of exceeding the novelty threshold after observing \( m > 1 \) samples is the integration of the EVD \( p_e(x) \),

\[
P(x > x_n) = \int_{x_n}^{\infty} p_e(x)\,dx = 1 - \int_{-\infty}^{x_n} p_e(x)\,dx = 1 - P_e(x) \tag{4.10}
\]

Figure 4.2 shows the probability that the novelty threshold (set at \( P = 0.99 \)) will be exceeded, \( P(x > x_n) \), for increasing numbers of samples \( m \). After \( m = 10 \) samples have been drawn, the probability that the \( P = 0.99 \) threshold will be exceeded is \( P(x > x_n) = 0.163 \), shown in the figure by the green line. After \( m = 100 \), \( P(x > x_n) = 0.797 \), shown by the blue line. By rearranging (4.8), we can determine that \( P(x > x_n) = 0.99 \) when \( m = 248 \). That is, after drawing \( m = 248 \) samples, our confidence that our novelty threshold (which we would normally expect to be broken 1% of the time) is actually exceeded with probability 99%, because we have drawn large numbers of consecutive samples from \( D \) (248 in this case).

Thus, setting a novelty threshold using the cdf \( P \) of the model of normality \( D \) is only valid when \( m = 1 \); i.e., for classification tasks in which a single entity is being compared with a model of normality. An example of this is when comparing a single mammogram to a model constructed using “normal” mammogram data (Tarassenko et al. 1995).

### 4.3 Continuous Monitoring

This thesis considers the case of continuous monitoring, in which we observe multiple consecutive samples from a jet engine. These samples may not be i.i.d., because there is a likely to be a correlation between the value of samples measured close together in time. Figure 4.3 illustrates the continuous monitoring case in which a window of \( m = 10 \) data is shown for an example process.

In this thesis, we will construct a model of normality, \( p(x) \), from a set of training data using the methods described in Chapter 3. We can then use EVT to determine where we expect the most extreme of \( m \) data points to lie with respect to that model under “normal” conditions, as described by the EVD, \( P_e(x) \). We will then compare windows of \( m \) test data to the EVD, to determine whether or not they lie within where we would expect the most extreme data to lie under “normal” conditions.
Figure 4.3: An illustration of data obtained during continuous monitoring, showing probability densities $p(x)$ for multivariate samples $x_i$ from an example process. Samples $x_4 \ldots x_{13}$ lie within the window shown by the grey background. The most extreme data point in this window is $x_5$ (circled), which occurs with the lowest probability density $p(x)$.

Figure 4.4: Extreme value distributions (EVDs) and cumulative distribution functions (cdfs) for $X \sim \mathcal{N}(0, 1)$, for increasing $m$. A novelty threshold set on the cdf $P_e(x|X) = 0.99$ (horizontal dashed line) for each value of $m$ is shown by a vertical dashed line.

The selection of the value of $m$ (i.e., the length of the window of data to consider) is a key question. EVDs for selected values of $m$ are shown in Figure 4.4(a). It can be seen that the majority of the probability mass described by the EVDs is located higher on the $x$-axis as $m$ increases, following the intuition that we expect higher maxima $x_{\text{max}}$ as more samples $X$ are drawn from $D$.

The cdf of the EVD, $P_e(x)$, describes the probability of observing a sample at $x$ if we draw $m$ samples from $D$. We have shown that setting a novelty threshold on the cdf of the model of normality $P = \kappa$, as described in Section 3.4, is not valid for continuous monitoring, and so we will instead set a novelty threshold on the cdf of the EVD, $P_e(x) = \kappa$.

Figure 4.5 shows that novelty thresholds, here set at $P_e(x) = \kappa = 0.99$, monotonically
increase along the $x$-axis with increasing $m$. That is, as we draw more samples from $D$, the location on the $x$-axis where we expect their maximum to fall with some set confidence $\kappa$ (the novelty threshold) increases.

Thus, the novelty threshold set at a constant probability of $P_e(x|\mathbf{X}) = \kappa$ corresponds to an increasing threshold in data space with increasing $m$. The choice of an appropriate value of $m$ will depend on the dynamics of the system being monitored, and the sampling rate.

### 4.4 Existing EVT Approaches

Previous work has used EVT to determine the position of novelty thresholds with respect to Gaussian distributions learned from a set of “normal” training data (Roberts 1999, Roberts 2000), according to the following steps:

1. Estimate the parameters of $D$, the underlying distribution representing “normality” from which it is assumed the normal training data have been generated. Maximum likelihood (ML) estimates of the parameters are made using the available training data.
2. Set novelty thresholds using the Gumbel distribution, as described in Section 4.1.1, where \( m \) is set to be the number of data in the training set, \( N \).

Consider an example training set of \( N = 10^2 \) normal data, \( \mathbf{X} \sim N(0,1) \), as shown in Figure 4.6(a). The EVD describing the upper tail of this distribution \( D \) has been computed using the two steps described above, where a novelty threshold has been placed at \( x_\kappa \approx 3.6 \), corresponding to \( P_e(x|\mathbf{X}) = 0.99 \). Note that the threshold has been set using the estimated distribution of the data, shown by the blue dashed line in the figure, which differs from their actual distribution, shown by the black line. The EVD, shown by the red line, is based on the estimated distribution, which causes the threshold to differ from the expected threshold of \( x \approx 4 \), as shown in Figure 4.4(b).

Observing \( N = 10^3 \) and \( N = 10^4 \) samples from \( D \), as shown in Figures 4.6(b) and (c), respectively, causes the novelty threshold at \( P_e(x|\mathbf{X}) = 0.99 \) to be placed at more extreme values, \( x_\kappa \approx 4.5 \) and \( x_\kappa \approx 4.9 \), respectively. These thresholds are close to the thresholds shown in Figure 4.4(b) because the estimated distribution of the data is close to their actual distribution for higher values of \( N \).

Note that the estimate of \( D \) is very close to the actual distribution \( D \) in both cases, yet the novelty threshold has been placed at more extreme values because we have observed more data \( N \), and used a method for which \( m \) is set to be equal to \( N \). Setting a novelty threshold further along the \( x \)-axis results in less sensitive novelty detection: \( x \) has to take ever more extreme values before it is classified “abnormal”.

As shown previously in Figure 4.5, the novelty threshold is placed at ever higher values on the \( x \)-axis, these values monotonically increasing with increasing EVT parameter \( m \), and where \( m \) increases with the increasing number of training data samples \( N \) for this method in which \( m = N \).

We argue that this behaviour is undesirable for continuous monitoring because, while observing more “normal” data should increase the accuracy in our estimate of the true parameters of \( D \) (that is, in the fit of the dashed blue line to the solid black line in Figure 4.6), it should not necessarily result in the novelty detection system becoming less sensitive. That is, as we observe more training data, our certainty in the model of normality should increase, but the novelty threshold should not necessarily increase without bound.
Figure 4.6: EVDs (shown in red) for increasing numbers of training data, computed using classical EVT methods. Though the underlying data density $D$ (shown in grey) is identical in each case, novelty thresholds (shown in green) are more conservative when greater numbers of data are observed. The estimate of $D$ is shown in blue for each case. Novelty thresholds are set at $P_e(x|X) = 0.99$.

For the problem considered by this thesis, in which data drawn from $D$ are observed sequentially (as described in Chapter 2), this disadvantage is readily apparent because the novelty threshold is set unboundedly higher on the $x$-axis (and thus novelty detection becomes unboundedly less sensitive) the longer that the system is observed. If we observe large numbers $N$ of training data and set the EVT parameter $m = N$, the novelty threshold would be set to such a high value on the $x$-axis that the resultant sensitivity of the model to abnormal data would be significantly reduced. Furthermore, an increase in the sampling rate will give rise to a higher value of $N$, and hence $m$. As a result, this method is not suitable for continuous monitoring applications.
4.5 Conclusions

We have introduced EVT, and shown that it can be used to determine the probability that a novelty threshold set on $P$ (as described in Chapter 3) will be exceeded. A novelty threshold set at $P = \kappa$ will be exceeded with probability $1 - \kappa$ if only one sample is observed. This assumption is valid for “single-instance” classification, such as in the case of classifying individual mammograms as either “normal” or “abnormal”. However, for novelty detection in which multiple data from one system are compared to a model of normality, as in the problem of continuous monitoring of jet engines considered by this thesis, the probability of exceeding the novelty threshold is greater than $P = 1 - \kappa$, and increases as more data are observed.

We used EVT to show that when a novelty threshold is set at $P = \kappa = 0.99$ for a model represented by a Gaussian distribution, that threshold will be exceeded with probability $P = 0.99$ after 248 data have been observed; i.e., after observing 248 data, the novelty threshold that should only be exceeded 1% of the time will in fact be exceeded 99% of the time. We conclude that we must use EVT to set the novelty threshold, such that the probability of threshold exceedance remains valid when multiple data are compared to the model of normality.

The novelty threshold defined by EVT describes where we expect the maximum of $m$ samples drawn from $D$ to occur under “normal” conditions. Thus, rather than set $m = N$, as with the existing method proposed by Roberts et al. (1998), we will set $m$ according to the length of the window of test data for the new EVT-based methods examined in this thesis, as shown in Figure 4.3.

Chapters 5 and 6 compare the application of the new EVT-based method for setting novelty thresholds with high-bandwidth tracked order data. The performance of the method is compared with that of the conventional method of setting a novelty threshold on the pdf, as described in Section 3.4.

Chapter 7 presents a novel high-bandwidth approach to novelty detection using full spectral data, in which there is no explicit EVT-based novelty threshold, though EVT is used to characterise the distribution of observation noise in vibration spectra. Chapters 8 and 9 compare the application of the new EVT-based method to low-bandwidth tracked order data with a heuristic technique currently used by engine experts.
Chapter 5

High-Bandwidth Novelty Detection

5.1 Introduction

As described in Chapter 2, the work described in thesis takes a high-bandwidth approach to monitoring development engines. Sample-by-sample novelty detection makes it possible to identify deterioration in engine condition as early as possible.

In the monitoring of development engines, a multivariate approach can be adopted, fusing information from multiple features, such as the amplitude of tracked order vibration and broadband spectral power. To allow sufficient degrees of freedom in the modelling approach so that the distribution of features during “normal” engine operation can be estimated accurately, we take a multimodal approach, using the GMM described in Chapter 3. Such multimodal approaches provide better estimates \( \hat{D} \) of the underlying distribution \( D \) assumed to have generated “normal” data than those provided by unimodal approaches (such as a multivariate Gaussian distribution).

We require a principled method of setting novelty thresholds for multivariate, multimodal pdfs, \( p(x) \). In Chapter 4, we described the use of EVT for setting novelty thresholds with univariate data. In order to apply this method to multivariate data, as will be described in Chapter 6, this chapter first describes a novel extension to EVT that accurately determines the EVD for multivariate, multimodal models of normality.

5.2 Multivariate EVT

5.2.1 Existing approaches to multivariate EVT

Existing work on multivariate EVT has concentrated on finding distributions to which multivariate extrema tend, analogous to the manner in which extrema for univariate
distributions tend towards one of the three known univariate EVDs (described in Chapter 4). The extrema considered by existing multivariate EVT may be defined as followed: given a set \( X = \{x_1, x_2, \ldots, x_N\} \) of \( N \) samples generated from some distribution \( p(x) \), where the \( i^{th} \) sample is \( Q \)-dimensional \( x_{i,Q} \), the \textit{componentwise-maximum} is defined (Resnick 1987) to be,

\[
M = (M_1, M_2, \ldots, M_Q)
\]

(5.1)

where \( M_j \) is the maximum of the \( j^{th} \) component (i.e., dimension) of \( X \):

\[
M_j = \max (x_{1,j}, x_{2,j}, \ldots, x_{N,j})
\]

(5.2)

Component-wise \textit{minima} are defined analogously.

Some of the original work on multivariate EVT was carried out by Hoeffding (1940) and Fréchet (1951), who investigated the convergence properties of multivariate maxima. Gumbel (1960) proposed the bivariate Gumbel distribution, but noted Fréchet’s result that “for given marginal distributions\(^1\) there exist infinitely many bivariate distributions with these margins.” Further groundwork on limit theory for multivariate extremes was provided by de Haan & Resnick (1977), while domains of attraction for various multivariate EVDs were considered by Marshall & Olkin (1983). Here, the authors proposed multivariate analogues to the univariate work of Gnedenko (1943), which described the domains of attraction for univariate EVDs.

Later methods involved estimation of the tails of the multivariate joint distribution, such as methods proposed by Tawn (1988) for bivariate joint distributions and Tawn (1990) for multivariate joint distributions. Tail estimation using ML methods is described by Smith et al. (1990). More recent work in this area has proposed tractable methods for estimating such tails using mixtures of Dirichlet distributions (Boldi & Davison 2007). Applications of tail-based estimation have included rainfall prediction (Coles & Tawn 1994), environmental pollution assessment (Heffernan & Tawn 2004), and tidal prediction (Butler et al. 2007).

The use of \textit{copulae} (Nelsen 1999) for modelling the interaction between marginal distributions in multivariate EVT has been investigated by Demarta (2002) and Demarta & McNeil (2005). Rather than finding the joint distribution, which becomes difficult for

\(^1\)i.e., the univariate distribution of a single dimension in the multivariate data, also referred to as “the margins” in the literature.
high-dimensional data, this method finds the “dependence structure” between dimensions. The copula is a multivariate function that forms a joint distribution by combining the separate marginal distributions using this dependence structure. This technique has proved popular in modelling financial extrema, with applications in modelling operational risk (Embrechts et al. 2001, di Clemente & Romano 2004, Dupuis & Jones 2006), credit risk (Frey et al. 2001), and large insurance claims (Cebrián et al. 2003).

Another approach to multivariate EVT is an extension of the peaks-over-threshold method (Coles & Tawn 1991, Ledford & Tawn 1996), which is a multivariate analogue of the generalised Pareto distribution (Castillo et al. 2005) in which excursions over some fixed (multivariate) threshold are modelled. Coles (2001) illustrates this technique using examples from financial and environmental modelling.

5.2.2 Using classical EVT for multivariate data

Classical EVT was first used for novelty detection in multivariate data by Roberts (1999) and Roberts (2000), with models of normality being represented by mixtures of Gaussian kernels. In multivariate space, the Gaussian kernel describes a hypersphere with $p(x)$ varying along a radius $r$ according to the univariate Gaussian (scaled by a normalisation factor dependent on dimensionality $Q$). That is, to determine the probability density $p(x)$ at any point in the hypersphere, the problem is reduced to a univariate case (in radius $r$). Roberts (1999) uses this assumption to reduce the problem of determining the EVD for a multivariate Gaussian kernel to a univariate case. As illustrated in Figure 5.1, the EVD for a single Gaussian kernel along a radius $r$ varies according to a univariate Gumbel distribution.

Thus, Roberts (1999) used the univariate equations of classical EVT to estimate Gumbel parameters $(c, d)$ for the EVD of the multivariate kernel. We note in passing that Roberts (1999) uses the following for classical EVT:

$$c_{2s} = \sqrt{2 \ln m} - \frac{\ln \ln m + \ln 2\pi}{2\sqrt{2 \ln m}} \quad (5.3)$$
$$d = \frac{1}{\sqrt{2\ln m}} \quad (5.4)$$

whereas Embrechts et al. (2008) uses $4\pi$ in the estimation of $c$, defined previously in (4.5). Thus, we denote the alternative use of $2\pi$ as shown above, to differentiate between the two approaches.
In order to investigate how this univariate (radial) EVD of a multivariate Gaussian kernel changes with dimensionality $Q$, we compare the estimates $(\hat{c}, \hat{d})$ obtained from classical EVT with those obtained experimentally by generating extrema using sampling. Finding estimates $(\hat{c}, \hat{d})$ from experimentally-generated extrema is performed by obtaining the maximum likelihood estimates (MLEs) of the Gumbel parameters. In both cases, $(\hat{c}, \hat{d})$ are estimates of the Gumbel parameters for the radial distribution of $D$ (using the univariate assumption, described above).

**Obtaining MLE Gumbel parameters**

The univariate Gumbel distribution is well-studied, and much work exists on the fitting of Gumbel distributions to univariate extrema. (Lowery & Nash 1970) compared four popular methods for estimating the parameters of the Gumbel distribution using a range of different data and concluded that MLE provided the closest estimate of actual parameters\(^2\).

---

\(^2\)in comparison to the method of moments, regression, and “Gumbel’s method”.
For independent extrema $X = \{x_1, x_2, \ldots, x_N\}$, we can write the likelihood

$$L(c, d) = p(y_1, y_2, \ldots, y_N|c, d) = \prod_{i=1}^{N} p(y_i|c, d) = \prod_{i=1}^{N} \frac{1}{d} \exp \left( -y_i - \exp(-y_i) \right)$$

where $y_i = (x_i - c)/d$ is the reduced variate, as before. We can thus write the log likelihood

$$\ln L(c, d) = -N \ln d - \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} \exp(-y_i)$$

which we wish to maximise by finding $(\hat{c}, \hat{d})$,

$$(\hat{c}, \hat{d}) = \arg\max_{c,d} \left( \ln L(c, d) \right)$$

Setting the partial derivatives of $\ln L(c, d)$ with respect to $c$ and $d$ to zero gives

$$P = m - \sum_{i=1}^{m} y_i + \sum_{i=1}^{m} y_i \exp(-y_i) = 0$$

$$Q = -m + \sum_{i=1}^{m} \exp(-y_i) = 0$$

as was first shown by Clarke (1973). We use an iterative method for producing estimates $(\hat{c}, \hat{d})$ as described in Algorithm 5.1 (Phien 1986), which rapidly converges to a maximum of $\ln L(c, d)$ (Worden et al. 2002).

For initial estimates $(c_0, d_0)$, we use approximations obtained from the equations of Gumbel moments expressed in terms of the sample mean and variance $(\mu, \sigma^2)$,

$$\mu \approx c + \gamma d$$

$$\sigma^2 \approx \frac{\pi^2 d^2}{6}$$

where rearrangement gives our initial estimates

$$d_0 = \frac{\sigma \sqrt{6}}{\pi}$$

$$c_0 = \mu - \gamma d_0$$

Figure 5.2(a) shows the distribution of $N = 10^6$ extrema generated from a bivariate Gaussian distribution $D = N(0, I)$, where each of the $N$ extrema is the maximum of a
Algorithm 5.1 MLE ($\hat{c}, \hat{d}$) for the Gumbel distribution

Provide initial estimates $(c_0, d_0)$

for iteration $i$ do

Compute $P, Q$

if $P, Q \approx 0$ then STOP

else compute

$$\Delta c = \frac{d_{i-1}}{N} \left[ \left( 1 + \frac{6(1-\gamma)^2}{\pi^2} \right) Q - \frac{6(1-\gamma)}{\pi^2} P \right]$$  \hspace{1cm} (5.10)

$$\Delta d = \frac{d_{i-1}}{N} \left[ \frac{6(1-\gamma)}{\pi^2} Q - \frac{6}{\pi^2} P \right]$$  \hspace{1cm} (5.11)

if $\Delta c, \Delta d \approx 0$ then STOP

else compute

$$c_i = c_{i-1} + \Delta c$$  \hspace{1cm} (5.12)

$$d_i = d_{i-1} + \Delta d$$  \hspace{1cm} (5.13)

end for

Figure 5.2: Fitting Gumbel distributions to extrema. Gumbel distributions with LSPRE and MLE parameters are shown in green and red, respectively. (a) Normalised histogram of $N = 10^6$ extrema, with fitted Gumbel distributions. (b) Extrema plotted on “Gumbel probability paper”, with similarly transformed Gumbel distributions.
set of $m = 100$ samples drawn from $D$. The Gumbel distribution with MLE parameters is shown in red, compared to a least-squares fit shown in green. Here, the latter was obtained by minimising the least-squares probability relative error (LSPRE), which was shown to be effective in Gumbel parameter estimation (Castillo 1988, Sohn et al. 2005), and which is defined to be

$$E = \sum_{i=1}^{N} w_i \left[ \hat{P}(x) - p(x|\hat{c}, \hat{d}) \right]^2$$

(5.18)

where $p(x|\hat{c}, \hat{d})$ is the value of the Gumbel distribution parameterised by estimates $(\hat{c}, \hat{d})$ at $x$, and $\hat{P}(x)$ is the empirical cdf formed by sorting $X$ in ascending order and defining $\hat{P}(x_i) = i/(N + 1)$. The weights $w_i$ are defined to be $w_i = 1/\hat{P}(x_i)$.

It may be seen from the figure that the MLE distribution is very close to the LSPRE distribution, yet the former converges to a solution much more rapidly than the latter.

Figure 5.2(b) shows the $N$ extrema transformed onto “Gumbel probability paper” (Castillo 1988) by transforming $\hat{P}(x) \rightarrow -\ln \left[ -\ln \hat{P}(x) \right]$. A straight line in the transformed space indicates a Gumbel distribution; the MLE and LSPRE distributions have been plotted in red and green, respectively. The transformed data lie close to these estimated distributions, yet diverge slightly in the tails. However, the fraction of data that lies close to the lines corresponding to the estimates is very high, as the data density is highest in the interval $x = [2.5 \ldots 4]$.

We use MLE parameters throughout the remainder of this chapter, because of the close fit of the resultant Gumbel distribution to the EVD of experimentally-obtained data, and the rapid convergence of MLE Algorithm 5.1 to a solution.

**Comparison of MLE parameters and those obtained using classical EVT**

We perform a series of Monte Carlo experiments to estimate the actual Gumbel parameters $(c, d)$ of the EVD, as described in Algorithm 5.2.

Thus, given $m$, we have $I = 100$ estimates $(\hat{c}, \hat{d})$, each obtained from a different set of $N = 10^6$ extrema. Figure 5.3 shows the ratio of the MLE Gumbel parameters $\hat{c}/\hat{d}$ evaluated over the experimental range of $m = 1 \ldots 10^4$. For each value of $m$, the mean and $3\sigma$ confidence intervals of the $I = 100$ experiments are plotted. This was performed for increasing dimensionality $Q = 1 \ldots 5$ of the Gaussian kernel. For comparison, the Gumbel parameters estimated by classical EVT are shown (where estimates obtained
Algorithm 5.2 Generate extrema and estimate $(\hat{c}, \hat{d})$, given $m$

\begin{verbatim}
for $I = 1$ to $100$ do
    /* For this $I^{th}$ experiment, find $N$ extrema */
    for $N = 1$ to $10^6$ do
        Draw $m$ random samples from $p(x)$
        Keep the most extreme of the $m$ samples
    end for
    Find $(\hat{c}, \hat{d})$ for the $N$ extrema
end for
\end{verbatim}

from both the methods of Roberts (1999) and Embrechts et al. (2008) are shown as red lines).

The figure shows that for a 1-dimensional Gaussian kernel, classical EVT closely estimates the values of $(c, d)$. While the method of Roberts (1999), given in (5.3), fits the observed data most closely for $m < 10^2$, the method of Embrechts et al. (2008), given in (4.5), appears to fit the data most closely for $m > 10^4$.

However, for dimensionality of the Gaussian kernel $Q > 1$, estimates obtained using either of the classical EVT methods do not fit the observed extrema. As $Q$ increases, the error of fit between the parameters estimated using classical EVT and those from observed extrema becomes greater.

An example of the discrepancy between the EVD of multivariate extrema and those predicted by classical EVT is shown in Figure 5.4. The standard bivariate Gaussian distribution $N(0, I)$ is shown in Figure 5.4(a), and a 2-dimensional histogram of $N = 10^6$ extrema generated from it (using Algorithm 5.2) is shown in Figure 5.4(b), where $m = 100$ in this example. This histogram is normalised such that it integrates to unity, and so it is an approximation $\tilde{p}_e(x)$ of the EVD.

Figure 5.4(c) and (d) show the EVDs parameterised by $(\hat{c}, \hat{d})$ obtained using classical EVT and MLE, respectively. While the MLE EVD closely matches the histogram of observed extrema, the EVD using parameters estimated with classical EVT is too small in radius.

Figure 5.5(a) shows, for the same standard bivariate Gaussian kernel, the mean-squared error (MSE) between $N = 10^6$ observed extrema and EVDs parameterised by varying combinations of $(c, d)$. The EVD using MLE parameters $\hat{c} = 3.04, \hat{d} = 0.32$ can be seen to correspond to the minimum error in the plot, showing that the MLE is a
Figure 5.3: Ratio of Gumbel parameters $c/d$ for increasing $m$. Ratios of parameters obtained using Classical EVT with $c_{2\pi}$ (5.3) and $c_{4\pi}$ (4.5) are shown by the upper and lower red lines, respectively. Ratios of experimentally obtained parameters for 1-, 2-, 3-, 4-, and 5-dimensional data are shown in black, blue, magenta, orange, and green, respectively. Experimentally-obtained parameters were obtained from 100 experiments, each experiment containing $10^6$ extrema, where each extremum was found by drawing $m$ samples from $N(0, I)$. For each value of $D$, mean $c/d$ ratios for the 100 experiments are plotted with solid lines; 3 standard deviations from the means are plotted with dashed lines.

close fit to the actual extrema. Parameter estimates obtained using classical EVT are $\hat{c} = 2.37$, $\hat{d} = 0.33$. This corresponds to a significant error between the resultant EVD and actual extrema, lying far from the location of minimum error in the plot.

The EVDs parameterised using estimates from classical EVT and MLE are compared against radii of actual extrema $r = \| \mathbf{x} - \mathbf{\mu} \|$ in Figure 5.5(b), where it may be seen that the EVD obtained using classical EVT significantly underestimates the radii of the distribution of the extrema.

The reason for this discrepancy can be determined by examining the distribution of probability mass in multivariate Gaussian kernels. Figure 5.6(a) shows a histogram of $p(x)$ values for $N = 10^6$ samples generated from a univariate Gaussian kernel with $m = 100$. The histogram is normalised such that it integrates to unity, which thus forms an empirical pdf over $p(x)$; i.e., the figure shows pdfs over densities, $p(p(x))$. It may be seen that the majority of probability mass occurs at higher values of $p(x)$.
Figure 5.4: Finding the EVD for a bivariate Gaussian distribution. (a) Bivariate Gaussian distribution, \( N(0, I) \). (b) Histogram of \( N = 10^6 \) extrema obtained by sampling \( p(x) \), where \( m = 100 \). (c) EVD with location and scale parameters estimated using classical EVT. (d) EVD with MLE parameters.

Figure 5.6(b)-(f) show similarly normalised histograms of \( p(x) \) for \( N = 10^6 \) samples generated from multivariate Gaussian distributions of dimensionality \( Q = 2, 3, 4, 7, \) and \( 12 \), respectively. As \( Q \) increases, probability mass tends to be concentrated close to densities \( p(x) = 0 \). As observed by Bishop (1995), though \( p(x) \) varies according to the Gaussian equation along a radius of the hypersphere, the probability mass tends to occur in a thin annulus of fixed radius.

Thus, as probability mass tends to cluster at increasing distances from the kernel with increasing \( Q \), so estimates obtained using classical EVT underestimate the radius of the EVD by increasingly large amounts. This is shown in the example in Figures 5.4 and 5.5, where the EVD obtained using classical EVT is of smaller radius than the
Figure 5.5: Determining \((c, d)\) parameters for the bivariate Gaussian kernel \(N(0, I)\). (a) Least squares error between \(N = 10^6\) extrema and \(p_e(x|c, d)\) for varying location and scale parameters \((c, d)\), where the minimum error occurs at \(c = 3.04, d = 0.32\). Classical EVT for \(m = 100\) gives \(\hat{c} = 2.37, \hat{d} = 0.33\). (b) Histogram showing radii (i.e., distance from kernel centre) of \(N = 10^6\) extrema, with EVDs estimated using classical EVT (red line) and MLE methods (green line).

distribution of observed extrema. It can also be seen in Figure 5.3 that the ratio of estimated parameters from classical EVT increasingly underestimates those of observed extrema for increasing \(Q\).

5.2.3 Determining EVD parameters for \(Q > 1\)

In determining the EVD \(p_e(x)\) for multivariate Gaussian distributions, we can use the univariate Gumbel distribution \(p_e(r)\) defined for radii \(r\)

\[
p_e(x) = \frac{1}{K} p_e(r|c, d)
\]

but we require three parameters: the Gumbel parameters \((c, d)\), which we obtain using the MLE method described in Section 5.2.2, and the normalisation constant\(^3\), \(K\). For a bivariate Gaussian distribution, we can find \(K\) by integrating the univariate EVD using polar coordinates:

\[
K = \int_{\theta=0}^{\theta=2\pi} \int_{r=-\infty}^{r=\infty} p_e(r|c, d) \ r \ dr \ d\theta
\]

\[
= 2\pi \int_{r=-\infty}^{r=\infty} p_e(r|c, d) \ r \ dr
\]

\(^3\)(Resnick 1987) aptly refers to the determining of a distribution’s normalisation coefficients as a “brutal business”. 

68
Figure 5.6: For multivariate Gaussian distributions, probability mass tends to be concentrated around $p(x) = 0$ as dimensionality $Q$ increases. Normalised histograms (integrating to unity) of $p(x)$ values are shown for $N = 10^6$ samples drawn from multivariate Gaussian distributions of $Q = 1, 2, 3, 4, 7, 12$ in (a)-(f), respectively.
Note that the resulting integral is the definition of the expected value (first-order moment) of \( p_e(r) \). For the Gumbel distribution, this is \( \mu_1 = (c + \gamma d) \), where \( \gamma \) is Euler’s constant. So,

\[
K = 2\pi \mu_1 \\
= 2\pi (c + \gamma d)
\]

which we can substitute into (5.19) to give the bivariate EVD:

\[
p_e(x) = \frac{1}{2\pi (c + \gamma d)} \left| \Sigma \right|^{1/2} p_e(r|c, d)
\]

(5.22)

Thus, we have formulated the EVD \( p_e(x) \) for a standard bivariate Gaussian distribution \( N(0, \textbf{I}) \) in terms of a scaled version of the univariate Gumbel distribution, varying with radius \( r \). In order to obtain the EVD for a bivariate Gaussian distribution with general covariance \( N(\mu, \Sigma) \), we apply a whitening\(^4\) transformation (Nairac et al. 1997) to \( x \) as follows:

\[
x' = \Lambda^{-1/2} \textbf{V}^T (x - \mu)
\]

(5.23)

where \( \Lambda \) is a diagonal matrix formed from the eigenvalues of \( \Sigma \), and \( \textbf{V} \) is the matrix of eigenvectors of \( \Sigma \). This is a transformation from Mahalanobis distance in \( x \) to Euclidean distance in \( x' \), from which radii may be found as before, and thus to which (5.19) may be applied to determine the EVD. We finally note that the EVD must be scaled according to the covariance matrix to ensure that it integrates to unity:

\[
p_e(x) = \frac{1}{2\pi (c + \gamma d) \left| \Sigma \right|^{1/2}} p_e(r|c, d)
\]

(5.24)

An example of the application of this method is shown in Figure 5.7. A histogram of \( N = 10^6 \) extrema, normalised such that it integrates to unity, is shown in Figure 5.7(a), where \( m = 100 \) in this case.

EVDs parameterised according to classical EVT and the method described above are shown in Figure 5.7(b) and (c), respectively. Again, the EVD obtained using classical EVT underestimates the radius of the actual EVD (with a mean-squared error of \( E = 0.03 \)), while the MLE method provides a close fit (with a mean-squared error \( E = 0.3 \times 10^{-4} \)).

While the above may be used when \( Q = 2 \), for models of dimensionality \( Q > 2 \), we must estimate \( K \) by integrating \( p_e(x) \) numerically, as described previously.

\(^4\)So called because the resultant eigenvalues of the data covariance matrix are transformed to unity, as with a white noise spectrum in signal processing terminology.
Figure 5.7: Finding the EVD for a bivariate Gaussian distribution with general covariance, $N(\mu, \Sigma)$, where $\Sigma = \begin{bmatrix} 4 & 1 \\ 1 & 0.7 \end{bmatrix}$. (a) Histogram of $N = 10^6$ extrema obtained by sampling $p(x)$, where $m = 100$. (b) EVD with location and scale parameters estimated using classical EVT. (c) EVD with MLE parameters.

5.3 EVT for Multimodal Models

5.3.1 Multimodal extrema

EVT defines an “extreme value” to be that which is either a minimum or maximum of a set\(^5\). This is due to the conventional use of EVT for determining extremely large or small events, such as extreme rainfall, extreme structural vibration, etc.

In novelty detection, when considering the extrema of unimodal distributions, as is

\(^5\)This requires that a partial order is defined for the set, as is typically the case for EVT in which variables are continuous and real. Demarta (2002) noted that there is no natural definition for extreme values of dimensionality $Q > 1$ since different concepts of ordering are possible.
Figure 5.8: Multimodal distributions require redefinition of the term “extreme value”. (a) In this bimodal distribution $p(x)$, extrema could fall between the two modes, and so we must consider more than “minimum” and “maximum” values on the $x$-axis. (b) In this multimodal Parzen windows model formed from $k = 60$ kernels, extrema could fall in the “horseshoe” between clusters of modes.

the focus of most previous work in the field of EVT, the existing definition is sufficient for univariate data. For multivariate data, providing that unimodal distributions are being considered, the existing definition may be taken to mean the minimum or maximum radius from the centre of the distribution (essentially reducing the multivariate case to a univariate case, varying in radius, as we have seen in Section 5.2.2).

However, for multimodal data, whether uni- or multivariate, the notion of minimum or maximum value is no longer sufficient, because there is no single mode from which “radius” may be defined. Figure 5.8(a) shows a univariate bimodal pdf. While minimum or maximum values in $x$ should be treated as extrema (e.g., $x = 1$ or $x = 28$), values between the two modes (such as $x = 10$) should similarly be taken to be extreme in terms of probability density $p(x)$, because they are just as improbable as the minimum or maximum on the $x$-axis.

This is a key point of departure for EVT in the context of novelty detection, in which we are interested in determining extremely unlikely events, whereas conventional EVT is interested in determining events of extremely large or small magnitude. As illustrated in Figure 5.8, while events of extremely large or small magnitude may be events that are extremely unlikely, not all events that are extremely unlikely are events of extremely large or small magnitude.

Similarly, Figure 5.8(b) shows a multivariate, multimodal distribution. For bivariate
samples \( x = (x_1, x_2) \), data points which represent a minimum or maximum in either dimension \( x_1 \) or \( x_2 \) should be considered “extreme”\(^6\), but it is also desirable that similarly improbable areas of data space, such as the origin \( x = (0, 0) \) in this example, should be considered “extreme” for the purposes of novelty detection.

As before, we require the EVD that describes where we expect the “most extreme” of \( m \) samples generated from the multimodal \( p(x) \) to lie. Thus, we redefine “extreme value” in terms of probability, as follows:

**Definition 1**

*For novelty detection, the “most extreme” of a set of \( m \) samples \( X = \{x_1, x_2, \ldots, x_m\} \) generated from a pdf \( p(x) \) is that which is most improbable with respect to the pdf; i.e., \( \arg\min_x p(x) \).*

Note that the conventional use of “extrema” to mean minimum or maximum values with respect to a unimodal distribution becomes a special case of the above definition: because probability density \( p(x) \) monotonically decreases with increasing distance from a single mode, selecting the samples furthest from the mode (the minimum and maximum of a set of samples) is also selecting the samples with the lowest \( p(x) \).

We note that this selection of extrema based on minimising \( p(x) \) is equivalent to selecting extrema by maximising \( P(\kappa) \), the probability volume contained by integrating the probability contour \( p(x) = \kappa \), because \( P(\kappa) \) monotonically increases as \( p(x) \) decreases.

Definition 1 is useful because it provides us with the mechanism we require to determine the extent of data space that is considered “normal”: if we observe \( m \) “normal” data generated from a model of normality, the EVD describes where the least probable of those \( m \) will lie. Thus, we can use the EVD to set a novelty threshold and perform novelty detection.

### 5.3.2 Previous approaches to multimodal EVT

As with multivariate EVT, little has been written concerning *multimodal* approaches to EVT with the exception of Roberts (1999) and Roberts (2000). In this approach, the multimodal pdf represented by a mixture of Gaussian kernels was reduced to a single-
kernel problem: to find the value of the EVD at some location $x$, the closest kernel (determined using Mahalanobis distance) was assumed to dominate $p(x)$, and thus the EVD was based on the Gumbel distribution corresponding to that closest kernel (using radius from the kernel centre, as described in Section 5.2.2). The contribution of other kernels to $p(x)$ was assumed to be negligible, and thus ignored. We hereafter refer to this as the “winner-takes-all” method\(^7\).

**Problems arising due to differing kernel variances**

Figure 5.9(a) shows the pdf $p(x)$, determined using the “winner-takes-all” method, corresponding to the univariate, bimodal distribution from Figure 5.8(a), which is a mixture of two Gaussian kernels. Here, the prior probabilities\(^8\) of each kernel are equal, $P(C_1) = P(C_2) = 0.5$, and the kernel variances are $\sigma_1^2 = 1$, $\sigma_2^2 = 4$. Figure 5.9(b) shows the corresponding cdf $P(x)$, from which it may be seen that each kernel is responsible for 0.5 of the total probability mass. The EVD modes for kernel $C_2$ have a maximum value of $p(x) = 0.15$, half of the maximum value of the modes for kernel $C_1$, because the modes for $C_2$ are spread out twice as far ($\sigma_2 = 2\sigma_1$).

In both Figure 5.9(a) and (b), the blue circles correspond to a histogram of $N$ extrema generated from each kernel independently. That is, kernel $C_1$ has been responsible for 0.5$N$ extrema and kernel $C_2$ has been responsible for 0.5$N$ extrema. This is not the same as drawing $N$ samples directly from the mixture $p(x)$, which is what we wish to perform.

Figure 5.9(c) shows the histogram obtained when $N$ extrema are generated correctly from the mixture; i.e., using Algorithm 5.2, in which $m$ samples are drawn from the mixture $p(x) = \sum_i P(C_i)p(x|C_i)$, and the most extreme is retained, for each of $N$ iterations. The cdf $P(x)$ in Figure 5.9 shows that kernel $C_2$ is responsible for more than 0.5 of the probability mass (in fact, two-thirds of it).

This example shows that, when samples are generated directly from the model of normality, kernel $C_2$, having wider variance and thus taking lower $p(x)$ values than kernel $C_1$, is responsible for more extrema than we would expect from the kernels’ equal prior probabilities. Though in each of $N$ iterations, we generate, on average,

---

\(^7\)using machine learning terminology in which only the closest kernel to a location $x$ in data space is used to determine network output; i.e., the kernel with the highest responsibility $P(C_i|x)$ for $x$.

\(^8\)alternatively known as mixing coefficients, or weights.
Figure 5.9: Comparison of EVD determined using the “winner-takes-all” method (red line) and histograms of samples (blue circles) drawn from bimodal distribution $p(x)$ (black line). (a) Samples are drawn from each kernel of $p(x)$ independently, according to their prior, $P(C_1) = P(C_2) = 0.5$, and so each kernel generates half of the samples. This closely matches the EVD determined by the “winner-takes-all” method, but does not represent the process of drawing $m$ samples from $p(x)$. (b) Corresponding cdf. (c) Extrema correctly drawn from the mixture $p(x)$ (i.e., $m$ samples are drawn from $p(x)$ and the most extreme is retained). The EVD determined by “winner-takes-all” method does not fit the actual observed extrema. (d) Corresponding cdf.
0.5m samples from each kernel (in accordance with their equal prior probabilities), we have a greater chance of retaining those samples from $C_2$ because the absolute values of their probability densities $p(x)$ are generally lower, due to the increased variance of that kernel. This unintuitive phenomenon was observed by Hyndman (1996) when considering conventional integration of mixtures of Gaussian kernels.

Thus, the assumption that only the closest kernel to $x$ need be considered when determining the EVD $p_e(x)$ cannot generally be made. Though the effect of other kernels on $p(x)$ may be negligible, because of their distance from $x$, their effect on $p_e(x)$ may be significant due to the relative differences in variances between kernels, as shown in the example in Figure 5.9. This typically occurs with Gaussian mixture models, in which kernels may have differing variances.

**Problems arising due to overlapping kernels**

The “winner-takes-all” method results in a piecewise-hyperspherical EVD, because $p_e(x)$ for all $x$ is determined using only a single kernel. This is illustrated in Figure 5.10, in which a mixture of two kernels with equal variance and significant overlap is shown. Figure 5.10(a) shows the actual EVD evaluated at $P_e(x) \leq 0.999$, evaluated using Monte Carlo methods. The resulting equiprobable contour follows the contours of the underlying pdf $p(x)$. Figure 5.10(b) shows the equivalent contour of the EVD obtained using the “winner-takes-all” method, in which the resultant contour is piecewise-circular. The difference between the two is shown in Figure 5.10(c), where it may be seen that areas of data space that should be considered “normal” (i.e., lying within the black line, if the contour is used as a novelty threshold) would be incorrectly considered “abnormal” by the “winner-takes-all” method.

The areas of data space thus misclassified as “abnormal” by the “winner-takes-all” method will increase with increasing overlap between kernels. For models of normality comprising large numbers of kernels, such as those constructed using Parzen windows estimation, the misclassified areas of data space could be large due to the typically considerable overlap between kernels and the typically large numbers of kernels used in such models.

We note that with increasing kernel dimensionality $Q$, the overlapped hypervolume of data space between neighbouring kernels (as a proportion of the total probability
Figure 5.10: A mixture of two kernels with equal variance and significant overlap. (a) The true EVD integrated to $P_e(x) \leq 0.999$ is shown by the black line. (b) The “winner-takes-all” method results in an equiprobable contour $P_e(x) \leq 0.999$ that is piecewise circular, shown by the red line. (c) Part of the overlapped regions of the kernels (between the red and black lines) would incorrectly be classified “abnormal” by the “winner-takes-all” method, using this contour as a novelty threshold.
mass) decreases. Thus, for high-dimensional models with smaller regions of overlapped data space, we expect that these problems will be less significant.

5.4 Understanding the EVD

As we have observed in Figure 5.10, the EVD for a distribution \( p(x) \) follows the probability contours of that distribution. This is a consequence of using Definition 1, where extrema are defined in terms of minimising \( p(x) \) for a set \( \mathbf{X} \) of \( m \) samples\(^9\).

Our definition satisfies the condition of Hyndman (1996), which states that the probability of observing data \( p(x) \) inside a novelty boundary (which he terms a “confidence interval”) should be at least as large as the probability of observing data outside the novelty boundary. That is, the novelty boundary is a lower bound on \( p(x) \) for “normal” data\(^10\).

It is convenient to consider the EVD as a transformation of equiprobable \( p(x) \) contours. Figure 5.11(a) illustrates this point, showing equiprobable contours for a model of normality represented by a mixture of three Gaussian kernels with general, non-spherical covariance. Figure 5.11(b) shows the model of normality, which itself can be considered to be a transformation of the equiprobable contours shown in Figure 5.11(a), in which each contour is weighted by its probability of occurring \( p(x) \) with respect to the model. Figure 5.11(c) shows the EVD for \( m = 100 \). Equiprobable contours of the EVD \( p_e(x) \) occur at contours \( p(x) \) of the model, and thus we may consider the EVD to be a weighting function of \( p(x) \) contours,

\[
p_e(x) = g[p(x)] \tag{5.25}
\]

for some weighting function \( g \). With the EVD defined in terms of \( p(x) \), we have the facility to accurately determine \( p_e(x) \) for complex, multimodal, multivariate distributions.

---

\(^9\)or, equivalently, maximising \( P(\kappa) \), the probability volume enclosed by the contour \( p(x) = \kappa \).

\(^10\)We note that though we are here defining a novelty threshold at some value of \( p(x) \), we are determining that value through integration of the EVD, the result of which will vary from model to model.
Figure 5.11: The EVD can be thought of as a weighting function acting on equiprobable $p(x)$ contours defined by the model of normality. (a) Equiprobable $p(x)$ contours constructed from a model of normality represented by a GMM of three kernels with general covariance. (b) The original generative pdf, with contours weighted proportionally to their probability density, $p(x)$. (c) The EVD corresponding to this model of normality for $m = 100$. The EVD operates on the same $p(x)$ contours, weighting each contour proportional to its extreme value probability density, $p_e(x)$.
Figure 5.12: Forming an EVD for a multimodal model of normality. (a) Normalised histogram of probability densities $p(x)$ obtained from $N = 10^6$ extrema ($m = 100$) drawn from a three-kernel GMM. (b) Constructing an EVD over the transformed probability densities, $p_e(\Psi)$. Gumbel and GEV pdfs are shown in green and red, respectively.

5.5 The $\Psi$-Transform

5.5.1 Modelling the distribution of $p(x)$ for extrema

Having observed that the EVD can be defined as a function operating on $p(x)$, we now require an estimate of that function. Figure 5.12(a) shows a normalised histogram of extrema generated from the mixture of three Gaussian kernels (from Figure 5.11) using Algorithm 5.2, for $m = 100$. The distribution is highly skewed towards $p(x) = 0$, as is expected for extrema. For a standard Gaussian kernel, $p(x) = (2\pi)^{-Q/2} \exp(-r^2/2)$, and thus $r = (-2 \ln p(x) - Q \ln 2\pi)^{1/2}$, allowing us to define a transform

$$\Psi \left[ p(x) \right] = \begin{cases} 
(-2 \ln p(x) - Q \ln 2\pi)^{1/2} & \text{if } p(x) < (2\pi)^{-Q/2} \\
0 & \text{if } p(x) \geq (2\pi)^{-Q/2}
\end{cases}$$

We note that if the original distribution were a single standard Gaussian distribution $N(\mu, \Sigma)$, the $\Psi$-transformation would map the $p(x)$ values back onto $r$, the radii of $x$ from $\mu$, which we know is distributed according to the Gumbel distribution (as shown in Section 5.2.2). The $\Psi$-transformation maps the distribution of $p(x)$ values back into a space into which a Gumbel distribution can be fitted, having observed that $p(x)$ for extrema are distributed approximately similarly for mixtures of negative exponentials of varying number of kernels, priors, and covariances. Subsequent sections in this chapter evaluate the $\Psi$-transformation with models of varying complexity and dimensionality.

Figure 5.12(b) shows the effect of the $\Psi$-transformation on $p(x)$ values obtained
from the examplar three-kernel mixture from Figure 5.11. The resultant distribution takes the Gumbel form, as shown by the green curve fitted to the transformed data. Here, we show the generalised extreme value (GEV) distribution fitted to the same data following the estimation methods of Park & Sohn (2006). The GEV was defined in (4.7), with the shape parameter $\xi \ll 0$ for Weibull distributions, $\xi \gg 0$ for Fréchet distributions, and $\xi = 0$ for Gumbel distributions. For the three-kernel examplar shown in the figure, $\xi = 0.06$, indicating that the distribution of these extrema is close to the Gumbel distribution\textsuperscript{11}.

**Bivariate mixtures**

Figure 5.13(a) shows the model of normality (shown previously in Figure 5.11), constructed using a mixture of three Gaussian kernels with general, non-spherical covariance. In this example, the kernels have unequal prior probabilities, $P(C_1) = 0.3$, $P(C_2) = 0.3$, $P(C_3) = 0.4$. Figure 5.13(b) shows a histogram of $N = 10^6$ extrema generated from the model of normality, with $m = 100$, normalised such that it integrates to unity. Thus, it is an estimate of the EVD, $\hat{p}_e(x)$.

The $\Psi$-transformed probability densities $\Psi[p(x)]$ for this model were shown previously in Figure 5.12(b), where we defined a Gumbel distribution over the transformed probability densities. Figure 5.13(c) and (d) show the EVDs obtained using the MLE of the Gumbel and general extreme value (GEV) distributions, respectively (as fitted in Figure 5.12). It can be seen that the difference between the Gumbel and GEV EVDs is negligible. The fit between each estimate of the EVD $\hat{p}_e(\Psi)$ and the EVD from experimentally-obtained extrema $\hat{p}_e(x)$ is very close (mean-square error $0.39 \times 10^{-4}$ and $0.38 \times 10^{-4}$ for Gumbel and GEV, respectively) showing that we have accurately determined the EVD for this mixture model.

**Parzen windows estimators**

Figure 5.8(b) showed a 60-kernel Parzen window estimator in which the distribution $p(x)$ covered a “U”-shaped area of data space. A normalised histogram of extrema generated from this model of normality is shown in Figure 5.14(a). The $\Psi$-transform method was used to estimate the EVD of this distribution for $m = 100$, again using $N = 10^6$.

\textsuperscript{11}We note that even when extrema are generated from a standard Gaussian kernel, which we know to be Gumbel distributed in its extremes, the estimation method employed typically returns $\xi \approx 0.06$. 
Figure 5.13: Finding the EVD of a multimodal model of normality. (a) Model of normality $p(x)$, represented using a three-kernel GMM with general covariance. (b) Estimate of EVD density $\tilde{p}_e(x)$ obtained from a histogram of $N = 10^6$ extrema for $m = 100$. (c),(d) Gumbel and GEV EVDs $\hat{p}_e(\Psi)$, respectively, obtained by forming a distribution over the transformed probability densities, $\Psi [p(x)]$.

extrema drawn from the model of normality. The $\Psi$-transformed probability densities are Gumbel distributed, and the resultant EVD is shown in Figure 5.14(b). Again, the fit between the resultant Gumbel distribution $\hat{p}_e(\Psi)$ is very close to the histogram of observed extrema (mean-squared error $0.9 \times 10^{-4}$).

**Mixture models for dimensionality $Q > 2$**

In order to examine the behaviour of the $\Psi$-transform for models of dimensionality $Q > 2$, a mixture of three Gaussian kernels of increasing dimensionality was constructed, with unequal prior probabilities $P(C_1) = 0.3$, $P(C_2) = 0.3$, $P(C_3) = 0.4$ and general covariance. For each model, $N = 10^6$ extrema were generated, their probability densities
Figure 5.14: Finding the EVD for the model of normality constructed using a Parzen window estimator with $k = 60$ kernels, as shown in Figure 5.8. (a) Histogram of $N = 10^6$ extrema drawn from $p(x)$, where $m = 100$. (b) Gumbel EVD obtained by forming a distribution over $\Psi [p(x)]$.

$p(x)$ determined and $\Psi$-transformed, and both Gumbel and GEV distributions fitted to the resultant $\Psi [p(x)]$ distribution.

Figure 5.15(a)-(d) show the resultant EVDs for models of dimensionality with dimensionality $Q = 3, 4, 5, and 6$. It can be seen in each case that the $\Psi$-transform causes the resultant $p(x)$ values of the extrema to be mapped onto a Gumbel distribution, and thus the proposed method of determining the EVD closely fits the observed extrema. In each case, we have shown the MLE GEV distribution, which is very close to the MLE Gumbel ($\xi \approx 0.06$ in each case).

We note that as dimensionality $Q$ increases, the mode $\mu_G$ of the Gumbel distribution increases on the $\Psi$-axis, with $\mu_G = 3.7, 3.9, 4.1, 4.2$ for $Q = 3, 4, 5, and 6$, respectively. This is to be anticipated from the results of Section 5.2.2, in which increasing dimensionality $Q$ causes probability mass to tend towards $p(x) = 0$. Note that the $\Psi$-transform is a negative-log transformation, which corresponds to probability mass tending towards higher values of $\Psi [p(x)]$, as shown in Figure 5.15.

We note that the curse of dimensionality does not seem to be problematic when considering distributions over transformations $\Psi [p(x)]$. Intuitively, we might expect that the number of samples $N$ required to accurately provide the histogram from which we perform the $\Psi$-transform should increase significantly with $Q$. That is, as dimensionality $Q$ increases, so $N$ might be expected to have to increase to sufficiently populate data space. However, in our experiments, the choice of value for $N$ provided that it is above

83
Figure 5.15: Finding the EVD for models of normality with dimensionality $Q > 2$. Each model is a three-kernel GMM with general covariance, for which the transformed probability densities $\Psi[p(x)]$ are shown. Gumbel and GEV EVDs are shown in green and red, respectively, defining the EVD over $\Psi$.

$N = 10^5$ has a negligible effect on the resultant values of the fitted distribution $(\hat{c}, \hat{d})$, independent of $Q$, for models of varying kernel numbers, priors, and covariances. This is because the problem has essentially been reduced to a univariate one, in which data are selected in univariate probability space, due to Definition 1, rather than in multivariate data space, and so the curse of dimensionality is not applicable.

Figure 5.16(a) shows a visualisation of the model of normality for $Q = 3$. This is a 4-dimensional visualisation, because every 3-dimensional location $x$ in data space is assigned a probability density $p(x)$, and so we show contour plots on three cross-sectional planes, where $p(x)$ is shown by colour.

A normalised histogram of extrema $\hat{p}_e(x)$ is shown in Figure 5.16(b), where again contour plots on a small number of cross-sectional planes are plotted. The Gumbel EVD
Figure 5.16: Visualising the EVD for a multimodal model of normality with $Q = 3$ dimensions. (a) Model of normality, $p(x)$, represented using a three-kernel GMM with general covariance. (b) Estimate of EVD density $\hat{p}_e(x)$ obtained from a histogram of $N = 10^6$ extrema for $m = 100$. (c) Gumbel EVD $\hat{p}_e(\Psi)$ obtained by forming a distribution over the transformed probability densities, $\Psi[p(x)]$. 

85
fitted to the $\Psi$-transformed probability densities, as shown fitted in Figure 5.15(a), is plotted as a cross-sectional contour plot in Figure 5.16. It can be seen that the Gumbel EVD is a close fit to the EVD of the observed extrema (mean-square error $0.39 \times 10^{-5}$). The corresponding MLE GEV distribution, not shown here, fits with $\xi = 0.06$ and mean-square error $0.39 \times 10^{-5}$, indicating that $\Psi[p(x)]$ of the extrema are distributed according to the Gumbel distribution.

5.6 Conclusions

5.6.1 Multivariate EVT

In order to perform multivariate novelty detection using jet engine vibration data, we require methods for setting novelty thresholds in multivariate space. We choose EVT to perform this, given its advantages described in Chapter 4.

We have reviewed existing approaches to using EVT with multivariate data, which mainly concentrate on finding minima and maxima in multidimensional spaces. Roberts (1999) used classical EVT for novelty detection, using mixtures of Gaussian kernels. The problem of multivariate analysis was simplified to univariate analysis by assuming that the EVD along a radius of a Gaussian kernel varies according to the Gumbel distribution. The parameters of the Gumbel were estimated using methods from classical EVT.

We have shown results of experiments in which it was shown that estimates of Gumbel parameters obtained using classical EVT, while closely modelling the distribution of univariate extrema, do not fit the distribution of multivariate extrema. We described how this is caused by the phenomenon in which probability mass tends to cluster around $p(x) = 0$ as dimensionality $Q$ increases.

The MLE method of obtaining Gumbel parameters was described, which results in EVDs that closely match observed extrema drawn from multivariate Gaussian distributions. For bivariate distributions, we found an expression for the normalising coefficient of the EVD in closed form, and also showed how the method is applicable to Gaussian distributions of general covariance.
5.6.2 Multimodal EVT

We showed that the conventional definition of “extreme value” based on the minimum or maximum of a set $X$ of $m$ samples is not applicable for multimodal models. We redefined “extrema” in terms of probability with respect to the model of normality $p(x)$, and showed that the conventional unimodal definition is a special case of our more general multimodal definition.

Existing approaches to using EVT for novelty detection with multimodal data are again those described by Roberts (1999) and Roberts (2000). In this work, the problem of multimodal analysis was simplified to unimodal analysis by assuming that the EVD at $x$ is dependent only on the closest kernel. It was assumed that only the closest kernel to $x$ contributes significantly to $p(x)$, and that the contribution from other kernels is negligible, and can thus be ignored.

When significant overlap between kernels exists, we showed that the piecewise-circular EVD obtained using previous methods (Roberts 1999, Roberts 2000) can fail to model the real EVD accurately. Also, the effect of other kernels on $p(x)$ is not negligible when there is significant kernel overlap (as typically occurs with a Parzen windows estimator).

Furthermore, we showed that, while the effect of other kernels may indeed be negligible on $p(x)$ if there is no significant overlap of kernels, it is not negligible on the EVD $p_{e}(x)$ if the variances of the kernels differ (as typically occurs with a Gaussian mixture model).

5.6.3 The EVD as a function of $p(x)$

We showed that the EVD can be considered to be a function operating on $p(x)$ contours, and redefined the EVD to be a distribution over a transformation of $p(x)$. We termed this the $\Psi$-transformation, and investigated its properties for Gaussian mixture models of general covariance and unequal prior probabilities, and for Parzen windows estimators with large numbers of kernels. We then showed that it provides a suitable model for the EVD in higher-dimensional models.

Thus, using the proposed method, we can provide a good estimate for the EVD for a model of normality of any dimensionality $Q$, assuming that the model is constructed
using some weighted mixture of Gaussian kernels. We note that the method should be applicable to mixtures of other distributions of decaying exponentials, such as those formed from Student’s $t$ (Svensen & Bishop 2005), gamma (Mayrose et al. 2005, Guillemot et al. 2005, Guillemot et al. 2006), and Weibull (Heckman & Singer 1984, Ishwaran 1996, Ebden et al. 2008) distributions, with small modifications to the $\Psi$-transform.

In the next chapter, we use the $\Psi$-transformation to set novelty thresholds for models of normality constructed using jet engine vibration data.
Chapter 6

High-Bandwidth Novelty Detection with Tracked Order Data

6.1 Introduction

This chapter describes the application of EVT techniques introduced in Chapter 4 to novelty detection in high-bandwidth engine vibration data. A multivariate, multimodal approach is taken using the extension to classical EVT described in Chapter 5. We present a method for automatic construction of models of normality using such data, as is required by an on-line engine monitoring system, and describe methods for data understanding such that the effectiveness of design choices can be evaluated graphically.

We compare results obtained retrospectively using conventional density-threshold methods (as described in Chapters 3 and 4) to results obtained using the new EVT-based method proposed in Chapter 5. We also compare results to those obtained using existing engine-mounted alarming systems.

6.2 Data

Table 6.1 lists the datasets considered by the work described by this chapter. All of these engines were developmental prototypes.

Table 6.1: Datasets used in the investigation described in this chapter.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Engine</th>
<th>Flights</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{A}_1$</td>
<td>3-shaft class A, engine I</td>
<td>8</td>
<td>Engine events due to loss of fan liner</td>
</tr>
<tr>
<td>$\mathcal{A}_2$</td>
<td>3-shaft class B, engine II</td>
<td>11</td>
<td>Exhibits blade failure</td>
</tr>
<tr>
<td>$\mathcal{A}_3$</td>
<td>3-shaft class A, engine II</td>
<td>7</td>
<td>Events caused by out-of-balance fan</td>
</tr>
<tr>
<td>$\mathcal{A}_4$</td>
<td>3-shaft class A, engine III</td>
<td>7</td>
<td>Damage caused by ice</td>
</tr>
</tbody>
</table>
6.2.1 Data description

Dataset $A_1$

This engine contains evidence of abnormal engine events in its final run, $A_{1,8}$, which were deemed to be caused by loss of the heat-resistant fan liner, due to the rubbing of an engine blade against the liner. Engine experts retrospectively determined that tracked-order data from runs $A_{1,5}$ and $A_{1,6}$ contain evidence of the fan rubbing that led to the eventual event. Conventional engine monitoring systems detected the final engine event in run $A_{1,8}$, because of the high level of vibration, but not the precursor episodes in runs $A_{1,5}$ and $A_{1,6}$.

Dataset $A_2$

This engine experienced a blade failure during run $A_{2,11}$, which conventional engine monitoring systems detected, again because of the high level of vibration. Retrospective analysis by engine experts showed that precursors to the event caused by minor engine damage were evident in data from runs $A_{2,7}$, $A_{2,9}$, and $A_{2,10}$, which were not detected by conventional systems.

Dataset $A_3$

A highly out-of-balance fan led to engine events during run $A_{3,7}$, and engine experts deemed that abnormal engine operation could be seen in data for the preceding run, $A_{3,6}$. Conventional monitoring systems did not detect either the final event, or the preceding abnormality.

Dataset $A_4$

This dataset was acquired during “ice testing”, in which the engine operates in conditions simulating icy weather. Engine experts deemed that engine damage was caused by ice, evident in data from runs $A_{4,5}$, $A_{4,6}$, and $A_{4,7}$. Again, conventional monitoring systems did not detect these events.

6.2.2 Train, validate, test

Figure 6.1 shows the partition of these datasets into those used for training/validation and those used for testing (using the approach described in Section 2.7). Noting that
the events contained in datasets $A_3$ and $A_4$ are more “subtle” than those in datasets $A_1$ and $A_2$ (because conventional monitoring systems did not detect events in the former due to the lower vibration magnitude), we use datasets $A_3$ and $A_4$ for testing, and the latter datasets for training/validation. This allows us to test the performance of resultant novelty detection systems on the more subtle classes of engine damage.

6.3 Methodology

6.3.1 Engine-specific novelty detection

Chapter 2 described the methodology adopted throughout the work described in this thesis, consisting of an adaption of the train-validate-test methodology for the purposes of engine-specific novelty detection. As detailed in Section 2.7, we determine the value of metaparameters (such as the probability at which to set a novelty threshold) based on results obtained from training and validation datasets. We then evaluate the resultant novelty detection scheme with these fixed values of the metaparameters using independent test datasets.

As described in Section 2.7, we take an engine-specific approach to novelty detection, and so models of normality $M(\theta)$ are constructed independently for each engine, and the parameters $\theta$ are engine-specific. As was shown in Figure 2.9, engine data from an engine run $r$ are compared to a model of normality constructed from all of the
Table 6.2: All features available in high-bandwidth datasets.

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5LP, 1LP, 1.5LP, 2LP, 3LP, 4LP, 5LP</td>
<td>Tracked orders for the LP shaft</td>
</tr>
<tr>
<td>0.5IP, 1IP, 1.5IP, 2IP, 3IP, 4IP, 5IP</td>
<td>Tracked orders for the IP shaft</td>
</tr>
<tr>
<td>0.5HP, 1HP, 1.5HP, 2HP, 3HP, 4HP, 5HP</td>
<td>Tracked orders for the HP shaft</td>
</tr>
<tr>
<td>BB</td>
<td>Total (broadband) spectral energy</td>
</tr>
<tr>
<td>RE</td>
<td>Residual (non-tracked order) spectral energy</td>
</tr>
</tbody>
</table>

previously-observed runs $1 \ldots r - 1$ that were deemed to be “normal”. If run $r$ is considered “normal” with respect to $M(\mathbf{\theta})$, data from run $r$ are added to the training set for that engine, and $M(\mathbf{\theta})$ is updated; otherwise, $M(\mathbf{\theta})$ remains unchanged and is used to test data subsequently observed during run $r + 1$.

We used this same approach throughout the work described in this chapter. However, further consideration must be given to the question of how a run should be classified as either “normal” or “abnormal” with respect to $M(\mathbf{\theta})$ for high-bandwidth applications. In such cases, each run may consist of very large quantities of data, each sample of which is individually classified as being either “normal” or “abnormal”. Thus, it would not be suitable to classify run $r$ as “abnormal” if any of its (potentially very numerous) constituent data were deemed “abnormal” with respect to $M(\mathbf{\theta})$. Noting that existing engine alerting systems consider engine events of greater than one minute duration to be sufficiently significant for raising an alert (Hayton et al. 2003, Clifton 2006), we adopt the rule that if data in an engine run are classified “abnormal” for a period of at least one minute, then the run is deemed “abnormal”, and the model $M(\mathbf{\theta})$ is not updated. This is analogous to the approach taken by Tarassenko et al. (2005), Tarassenko et al. (2006), and Hann (2008) in novelty detection using human vital-sign data, in which the corresponding above-threshold time interval required to generate an alarm is related to standard practice in the care of in-hospital patients.

We assume that the first run $r = 1$ in a dataset is normal, such that $M(\mathbf{\theta})$ can be constructed using data from that run.

6.3.2 Selecting the dimensionality of multivariate models

Chapter 2 described features typically used by engine experts to determine engine condition, such as tracked orders. Chapter 5 described a multivariate approach to novelty
detection, in which the joint distributions of many such features can be modelled, rather than being considered independently. This approach offers the advantage that subtle relationships between features may be learned, rather than considering only the absolute magnitude of individual features.

The use of available features in our multivariate approach will now be considered. We could construct a single model of normality using feature vectors comprised of all available features. Table 6.2 lists all of the features available in the datasets used by the work described by this chapter\(^1\).

It may be seen from the table that 7 tracked orders are monitored for each shaft (where we typically consider the vibration magnitude of each tracked order, as described in Chapter 2). Broadband and residual energy, BB and RE, are not associated with any shaft, by definition. Thus, a model describing the joint distribution of all available features would be 23-dimensional in this case.

However, abnormal system condition is not likely to be characterised by simultaneous changes in many features. For example, a fault in one engine shaft could be expected to change the characteristics of some of the tracked orders associated with that shaft, and perhaps the shaft-independent features (BB and RE), but it would be unlikely to change the characteristics of features associated with other shafts because there is no mechanical coupling between shafts. Thus, of the 23 dimensions in our example, we might expect 9 or fewer (7 tracked orders + BB + RE) to exhibit changes associated with an abnormal condition in an engine shaft. In practice, engine abnormalities result in changes in a small number of tracked orders, typically 5 or fewer (Clifton et al. 2007, King et al. 2009). In such cases, the majority of dimensions in the model would take “normal” values, even during periods of engine abnormality, and thus the output of the model would be biased towards normality.

A compromise exists between making the model sufficiently descriptive (i.e., of sufficiently high dimension) such that any engine abnormality can be identified, and making it sufficiently parsimonious (i.e., low-dimensional) such that the problem of being biased towards “normality” is avoided. Automatic methods have been proposed for the selection of features (Siedlecki & Sklansky 1988, Jain & Zongker 1997), but for engine health

\(^1\)Residual energy is that residual of the broadband spectral energy not associated with the tracked orders; i.e., \(RE = BB - (0.5LP + 0.5IP + 0.5HP + 1LP + 1IP + \ldots)\)
monitoring, prior knowledge suggests that we group features into separate models for each shaft as follows:

- **LP-shaft model**: $[0.5LP \ 1LP \ 1.5LP \ 2LP \ 3LP \ 4LP \ 5LP \ BB \ RE]$
- **IP-shaft model**: $[0.5IP \ 1IP \ 1.5IP \ 2IP \ 3IP \ 4IP \ 5IP \ BB \ RE]$
- **HP-shaft model**: $[0.5HP \ 1HP \ 1.5HP \ 2HP \ 3HP \ 4HP \ 5HP \ BB \ RE]$

where each model is 9-dimensional for the 3-shaft engines considered in this chapter (as described in Section 6.2).

Figure 6.2 shows a flowchart of the steps described in subsequent sections that process and model the data for each of the three shaft-based models listed above.

### 6.3.3 Speed-banded models

In addition to the available features listed in Table 6.2, engine shaft speeds are also measured. Engine vibration increases in magnitude and variability in a non-linear manner with respect to increasing shaft speed (Nairac et al. 1997, Nairac et al. 1999, King et al. 2009). In order to consider separately the distribution of engine vibration at different speeds in the speed range, we divide the speed range $[\omega_{idle} \ 100\%]$ into four equal sub-intervals (*speed-bands*), where $\omega_{idle}$ is the shaft speed at which the engine is idle. We denote these four speed-bands $S_1 \ldots S_4$, for increasing shaft speed. Thus, data are divided between the four speed-bands according to the speed of the engine at the time...
Figure 6.3: An example time-series of engine shaft speeds $\omega$, in which the engine speed is held at approximately 92% max. shaft speed, before being decelerated and accelerated between 60% and 90% max. speed. The four speed-bands $S_1 \ldots S_4$ are shown by the grey shaded background, where $S_1$ is the lowest speed-band and $S_4$ is the highest.

of sampling.

Figure 6.3 shows an example, in which the exemplar engine maintains an engine speed of 90% for most of the period shown, within speed-band $S_4$. Near the end of the period shown, a series of sudden engine decelerations and accelerations are performed, moving between the sub-ranges of speeds covered by speed-bands $S_1$ to $S_3$.

For each shaft, we construct a separate model for each of the four speed-bands, using training data that occur while the engine is operating within that speed-band. During the testing phase described in Section 6.3.1, test data are compared to the appropriate speed-band model depending on the current engine speed.

### 6.3.4 Pre-processing

In order to prepare the data for analysis, pre-processing must be performed, as described in Section 2.3.

**Filtering and normalisation**

To remove transient single-sample noise, a median filter of window length $W = 3$ was applied to each of the features listed in Table 6.2. Component-wise normalisation was then performed for all features, such that each varied over approximately the same range, making the analysis independent of the absolute values of each feature $x$:

$$x' = \frac{x - \mu}{\sigma}$$

(6.1)
where $\mu$ and $\sigma$ are the mean and standard deviation, respectively, of feature $x$ in the training data. This normalisation is alternatively described as *standardisation* in the literature (Scott 1992), because it would transform a Gaussian distribution $N(\mu, \sigma)$ into the standard Gaussian distribution $N(0, 1)$.

If feature $x$ was initially distributed according to the Gaussian distribution, for example, then approximately 99.7% of the resulting normalised data $x'$ would be expected to lie within $3\sigma$ of the mean$^2$. However, it is likely that some features are not distributed according to the Gaussian distribution, in which case (6.1) may simply be considered to be a linear transformation (a shift by some constant $\mu$, followed by a scaling by some constant $\sigma$) that causes each feature to vary over approximately the same normalised range.

**Data reduction**

When building a dataset for training a data-driven model, it is important that data are acquired as uniformly as possible over the entire operating range of the system. For example, an in-service jet engine will spend much of its time operating at steady-state “cruise” conditions, with much less time spent operating under take-off or landing conditions, while a developmental engine may spend much of its time operating at steady-state conditions between testing manoeuvres. If training data were drawn randomly from the

$^2$obtained by integrating $p(x) = N(0, 1)$ over the range $x = [\mu - 3\sigma \quad \mu + 3\sigma]$
entire run, then this would result in a strong bias towards steady-state conditions, simply because there would be many more feature vectors derived from periods of steady-state operation.

A balanced dataset may be constructed by discarding feature vectors during steady-state conditions if the change in engine speed from the previous feature vector is below a given threshold. We discard a sample \( x_t \) occurring at time \( t \) with engine speed \( \omega_t \) if \( |\omega_t - \omega_{t-1}| < \omega_z \), for some speed-based rejection threshold \( \omega_z \).

Figure 6.4 shows the proportion of data retained in datasets \( A_1 \ldots A_4 \) for varying speed-based rejection threshold \( \omega_z \) between 0% and 0.1% max. shaft speed, and after discarding feature vectors that contain “invalid” values. The latter are feature vectors observed when the engine is operating at sub-idle speeds, \( \omega < \omega_{\text{idle}} \), or feature vectors that contain tracked order vibration amplitudes that are artefactually negative (as occurs when the acquisition system described in Section 2.4 cannot adequately locate a peak in the vibration spectrum corresponding to a tracked order). It may be seen from the black line in the figure that discarding “invalid” feature vectors causes the quantity of retained data to fall to approximately 70% of the original amount.

It may also be seen that increasing \( \omega_z \) from 0% to 0.01% causes the amount of retained data to decrease from a maximum of approximately 70% of the original data quantity to approximately 17%. Subsequent increases in the value of \( \omega_z \) cause asymptotically smaller quantities of data to be discarded. This indicates that steady-state conditions exist frequently throughout datasets \( A_1 \ldots A_4 \). This is expected, because the data used for the investigation described by this chapter were acquired from developmental engines which are operated at steady-state conditions for long periods (as shown by the example in Figure 6.3).

An example of the effect of using speed-based rejection is shown in Figure 6.5, where the time-series of engine shaft speeds for a period of operation from dataset \( A_{1,1} \) is shown before and after application of this pre-processing step, with \( \omega_z = 0.05\% \). Before speed-based rejection (shown in the upper plot), large quantities of data may be seen to be observed at approximately equal shaft speeds, particularly during the initial period of steady-state behaviour at \( \omega_{\text{IP}} \approx 59\% \) occurring at time \( t \) between 2065 and 2075, and during the final period of steady-state behaviour at \( \omega_{\text{IP}} \approx 57.5\% \) occurring at time \( t \).
Figure 6.5: An example of speed-based data rejection, showing $\omega_{IP}$ for data from an episode in dataset $A_{1,1}$. The upper subplot shows the original series of data, and the lower subplot shows the same period after speed-based rejection has been applied such that a sample is discarded if its associated shaft speed lies within $\pm 0.05\%$ of the shaft speed associated with the previous sample.

Figure 6.6: Number of data in speed bands $S_1 \ldots S_4$ of dataset $A_{1,1}$ before and after speed-based rejection (shown in the upper and lower plots, respectively), using speed threshold $\omega_z = 0.05\%$. 

98
between 2090 and 2100. After application of speed-based rejection (shown in the lower plot), most data observed during these steady-state periods have been discarded, while the number of data points occurring during the deceleration from time $t = 2078$ to $t = 2090$ has been reduced.

Figure 6.6 shows the number of data points observed within each speed-band for run $A_{1,1}$ before and after speed-based rejection, with $\omega_z = 0.05\%$. During this run, in which the engine was left idling for long periods, it may be seen that the majority of data were observed within speed-band $S_1$, which has been significantly reduced by speed-based rejection. After application of speed-based rejection, the number of data observed within each speed-band is approximately similar.

### 6.3.5 Visualisation

In order to investigate the effects of pre-processing, and to provide further understanding of structure in the high-dimensional datasets, visualisation of the 9-dimensional feature vectors acquired in each run is performed.

The process of visualisation maps feature vectors from their original $Q$ dimensions (data space) into a 2- (or 3-) dimensional space (latent space) for visualisation, such that each vector is mapped onto a corresponding single point in this visualisation space. Feature vectors similar to one another in data space map to points that are close to one another in latent space while feature vectors that are significantly far apart in data space should map to points that are far apart in latent space. As a result, groups of similar feature vectors form clusters in the visualisation.

In datasets suitable for the application of novelty detection techniques, feature vectors that correspond to “normal” data (which are often similar) form such clusters. Feature vectors that correspond to “abnormal” data (which are dissimilar to “normal” feature vectors) appear as outlying points, away from the normal clusters. Thus, we can determine qualitatively the effect of selecting various features for inclusion in the feature vectors by examining the separation of “normal” and “abnormal” data in the visualisation, with the goal of selecting features that result in maximum separation.

Similarly, the effects of various pre-processing methods can be assessed using visualisation, again with the goal of choosing methods that maximise separation between “normal” and “abnormal” data, such that abnormal data may be clearly identified.
Finally, 2-D visualisation may be useful in conveying the results of high-dimensional analysis to a user during monitoring of the system. Visualisation can provide graphical explanation of decisions made by the monitoring system, increasing user confidence and allowing a more intuitive interpretation of system state (Clifton et al. 2008).

**Sammon’s mapping**

Sammon’s mapping (Sammon 1969) is a transformation that attempts to best preserve, in the low-dimensional latent space, distances between feature vectors in their original data space. The Sammon stress metric $E_{\text{Sam}}$ is defined using the distance $d_{i,j}$ between pairs of the $N$ feature vectors $(x_i, x_j)$ in data space, and the distance $d_{i,j}^*$ between the corresponding pair of projected feature vectors $(y_i, y_j)$ in latent space:

$$E_{\text{Sam}} = \frac{1}{\sum_{i=1}^{N} \sum_{j=i}^{N} d_{i,j}^*} \sum_{i=1}^{N} \sum_{j>i}^{N} \frac{(d_{i,j} - d_{i,j}^*)^2}{d_{i,j}}$$

The distance used is typically Euclidean, and the optimisation is typically performed using gradient descent (Nabney 2002).

A disadvantage of using Sammon’s mapping for projection of data is that the mapping is only defined for those $N$ feature vectors used in its construction, whereas we may wish to visualise data not originally included in the construction of the mapping. Attempts have been made to generalise the mapping to include previously-unseen feature vectors (Pekalska et al. 1999).
NeuroScale

The NeuroScale training algorithm (Lowe & Tipping 1996) seeks to minimise the error function $E_{\text{sam}}$ in order to best preserve the original inter-point distances $d_{i,j}$ after mapping the points into latent space. The mapping is learned using a radial basis function (RBF) neural network with $Q$ inputs, $P = 2$ output nodes, and a single hidden layer of $H$ hidden nodes, as shown in Figure 6.7.

Each of the $H$ nodes in the hidden layer corresponds to a radial basis function in $Q$-dimensional data space. $H$ is typically selected to be an order of magnitude greater than the dimensionality of the feature vectors $Q$, and an order of magnitude smaller than the number of feature vectors $N$. For example, if $Q = 10$ and $N = 1000$, an appropriate number of hidden nodes would be $H = 100$.

To construct the NeuroScale mapping, the parameters of the $H$ radial basis functions and the values of the output weights (i.e., those from the hidden layer to the output layer) must be determined. To achieve this, the neural network is first initialised, and then trained using the training set of $N$ feature vectors.

1. **Initialisation:** The initial centres of the $H$ radial basis functions are set to be the $Q$-dimensional co-ordinates of $H$ feature vectors randomly selected from the training set. There are a number of algorithms which may be used for setting the initial width of the $H$ radial basis functions (Lowe & Tipping 1996, Nabney 2002). Typically, the width is set to be the distance between the two centres which are furthest apart in order to achieve good coverage of data space.

The output weights from the $H$ hidden nodes to the $P$ output nodes are initialised using Principal Component Analysis (Bishop 2006). The eigenvectors of the training set covariance matrix with the largest eigenvalues are found. These principal components are the $P$ vectors in data space that account for maximum variance in the data, and they are used as the initial values for the target outputs from which the initial values of the output weights are computed (Bishop 1995).

2. **Training:** Training the RBF network is a two-stage process. In the first stage, the parameters of the radial basis functions are set so that they approximately model the probability density of the training set. In the second stage, the output
weights are learned by optimising the Sammon stress metric using methods from linear algebra (Nabney 2002).

Once training is complete, feature vectors may be visualised by presenting them at the input layer to the NeuroScale network, which then gives the corresponding 2-dimensional co-ordinates of the mapped feature vector at its output.

In order to visualise the datasets used in the investigations described in this chapter, we reduce the number of data in each dataset using $k$-means clustering, such that the NeuroScale network can be trained. Typically, this requires reducing the dataset to approximately $k = 500$ cluster centres in order for the training procedure to be performed\footnote{Recent work has been proposed that shows it may be possible to visualise large datasets in which $N \gg 500$ (Wong et al. 2008), and which could avoid the need to summarise the dataset using $k$-means clustering.}. With $Q = 9$ dimensional feature vectors, and $k = 500$ training data, we select $H = 50$ hidden nodes in the NeuroScale architecture, following the rule-of-thumb described above.

Figure 6.8(a) shows the result of visualising all data in speed-band $S_3$ (after pre-processing) in datasets $\mathcal{A}_{1,1} \ldots \mathcal{A}_{1,3}$. The NeuroScale network was trained using the method described above, after reducing the data to $k = 500$ cluster centres using $k$-means clustering. These $k$ cluster centres are shown projected in 2-D as black crosses in the figure. Unlike Sammon’s mapping, once the NeuroScale network learns the mapping
between data space and latent space, the original \( N \) data (where \( N \) is much greater than the number of data \( k \) used to train the network) may then be projected using the trained network. The figure shows the original data projected into 2-D as grey dots. It may be seen that the \( k = 500 \) cluster centres approximate the distribution of the original data in the projection, and hence should approximate the distribution of the original data in \( Q \)-dimensional data space.

Figure 6.8(b) shows a projection of the original data before and after pre-processing (including speed-based rejection). It may be seen that the distribution of pre-processed data (shown in grey) accurately represents the distribution of original, unprocessed data (shown in black), and thus we conclude that the use of speed-based rejection has not resulted in a significant loss of information.

We will present results using NeuroScale visualisation throughout the remainder of this chapter.

### 6.3.6 Model construction

Following pre-processing, a model of normality \( M(\theta) \) may now be constructed using the training data available for each of the speed-bands. We choose to use a GMM, and hence the number of kernel centres \( K \) in the mixture must be determined. As described in Section 2.7, \( K \) should be selected so as to model the distribution \( D \) from which the “normal” training data are assumed to have been generated, while not over-fitting the training data so as to perform poorly with previously-unseen data. In this thesis, we assume that the underlying distribution \( D \) will vary between engines (as described in Chapter 2), and so we require a method for automatically determining the optimal value of \( K \) given some training data from a specific engine.

**Determining the number of free model parameters**

The selection of the value of \( K \) can be performed automatically using *cross-validation*, whereby all available data are partitioned into \( P \) subsets of equal size, and an overall measure of fitness (the negative log likelihood, \(-\ln \mathcal{L}\)) is computed as described in Algorithm 6.1.

In practice, whenever a GMM is required to be fitted to data, it is conventional to construct a number of candidate models in parallel, using the same training data, and
select that model which best fits the data. This is necessary because the EM algorithm, as described in Section 3.3.3, is non-deterministic and can result in models with varying degrees of fit to the training data. Thus, we train 10 candidate GMMs in parallel and select that which best fits the dataset $X$.

For illustrative purposes, Figure 6.9 shows the negative log likelihood during training of candidate GMMs with $K = 3$ kernels in Step 1 of Algorithm 6.1, for iteration $p = 1$. (The choice of a suitable value of $K$ is described below.) The number of EM iterations

\begin{algorithm}
\textbf{Algorithm 6.1} $P$-fold cross-validation

\small

\begin{algorithmic}
  \State Partition $N$ training data into $P$ subsets, $X_i$, $i = 1 \ldots P$
  \For{$p = 1$ to $P$} \Do
    \State 1. Fit 10 candidate GMMs to the union of subsets $\bigcup_{i \neq p} \{X_i\}$, using EM (i.e., omitting the $p$th subset from the training set)
    \State 2. Using the $p$th subset as a “validation” set for this iteration, find the negative log likelihood $-\ln \mathcal{L}(X_p)$ w.r.t. each of the 10 GMMs
    \State 3. Find $-\ln \mathcal{L}_p$, the average of the 10 negative log likelihoods for iteration $p$
  \EndFor

\end{algorithmic}

Find the average negative log likelihood $-\ln \bar{\mathcal{L}}$ over the $P$ iterations,

$$-\ln \bar{\mathcal{L}} = \frac{1}{P} \sum_{p=1}^{P} -\ln \mathcal{L}_p$$  \hspace{1cm} (6.3)
during training of these 10 candidate GMMs is shown on the x-axis. It may be seen that the negative log likelihood of the training data with respect to each GMM decreases rapidly during the initial EM iterations, then reaches a constant value. The model corresponding to the lower red line results in the best fit to the data, with $-\ln L \approx 2.34 \times 10^4$ after 50 iterations of the EM algorithm.

Continuing our illustration, Figure 6.10 shows the negative log likelihood $-\ln \bar{L}$ averaged over $P = 5$ iterations of 5-fold cross-validation, where Algorithm 6.1 has been used for each value of $K$ in the range $1 \ldots 20$. This has been repeated for each speed-band, $S_1 \ldots S_4$. The figure shows that the negative log likelihood $-\ln \bar{L}$ decreases as $K$ is increased from $K = 1$, reaches a minimum between $K = 3$ and $K = 5$, and then rises as $K$ is further increased.

This is to be expected: a model with too few free parameters ($K$, in this case) will not accurately represent the underlying distribution $D$, and so will fit validation data poorly. At some optimal value for the number of free parameters, the model will accurately estimate $D$ using the training data, and will thus optimally fit the validation data. Increasing the number of free parameters past this optimum will cause the model to over-fit the training data, and perform increasingly poorly with validation data.

The figure highlights the “optimum” value of $K$ for each of the speed-bands with a circle, showing how model complexity can be automatically determined using this method.

**Constructing a final model of normality**

With the “optimal” value of $K$ selected for a given training set, for each speed-band, a final model of normality $M(\theta)$ may now be constructed using all available training data in each speed-band; i.e., from $\bigcup_{i=1}^{P} \{X_i\}$. 10 candidate “final” GMMs were constructed using the “optimal” value of $K$ selected by cross-validation.

Figure 6.11(a) shows the negative log likelihood $-\ln L$ of all training data with respect to the best-fitting candidate of the 10 final GMMs as a thick black line, during 20 EM iterations. For comparison, the values of $-\ln L$ for the best-fitting GMMs in each of the $P = 5$ folds of the cross-validation algorithm are shown by thin lines. It may be seen that the $-\ln L$ value of the best-fitting candidate final GMM is significantly higher ($-\ln L \approx 2.9 \times 10^4$) than the value for those models trained during cross-validation.
Figure 6.10: Selecting the optimal complexity of parameters $\theta$ in model $M(\theta)$. For each of the four speed bands, 5-fold cross-validation was performed 10 times. The negative log likelihood $-\ln L$ was calculated for each of the 10 cross-validations, and averaged. The figure shows $-\ln L$ averaged for 10 cross-validations using data from datasets $A_1 \ldots A_3$, for speed bands $S_1, S_2, S_3$, and $S_4$ shown in black, blue, red, and green respectively. The minimum averaged log likelihood gives the optimal model complexity for each speed band, shown by circles.

Figure 6.11: For each of the 5 folds in the 5-fold cross-validation, the negative log likelihood $-\ln L$ for the GMM of best fit is shown throughout the 20 training iterations (plotted using thin lines). The negative log likelihood for the GMM of “optimal” complexity trained using data from all 5 folds is plotted as a thick line. (a) With unnormalised $-\ln L$, the “optimal” GMM has higher overall error than the best-fit GMMs for each of the 5 folds. (b) With normalised $-\ln L$, the error of the “optimal” GMM may be seen to be similar to the errors of the best-fit GMMs for each of the 5 folds. Data from speed band $S_3$ using datasets $A_1 \ldots A_3$ were used in this example.
Figure 6.12: Empirical probability distribution of negative log likelihood $-\ln \mathcal{L}_{\text{norm}}$ for all GMMs used during 5-fold cross-validation, using datasets $A_1 \ldots A_3$.

$(-\ln \mathcal{L} \approx 2.3 \times 10^4)$. However, this is because the best-fitting final GMM is trained using data from all $P = 5$ subsets, while those from cross-validation are each trained using data from $P - 1$ subsets. We define a normalised negative log likelihood

$$-\ln \mathcal{L}_{\text{norm}} = \frac{-\ln \mathcal{L}}{N}$$

(6.4)

where $N$ is the number of data in the training set used to determine $-\ln \mathcal{L}$.

Figure 6.11(b) shows $-\ln \mathcal{L}_{\text{norm}}$ for the GMMs considered previously, where it may be seen that $-\ln \mathcal{L}_{\text{norm}}$ for the best-fitting final GMM (shown by the thick black line) is now similar to those of the GMMs from the 5 folds of the cross-validation, at $-\ln \mathcal{L}_{\text{norm}} \approx 7.6$, allowing us to conclude that the final GMM is as good a fit to the full training data as the models considered during cross-validation.

This conclusion is supported by the histogram shown in Figure 6.12, which shows the distribution of $-\ln \mathcal{L}_{\text{norm}}$ for each of the 10 candidate GMMs trained during the $P = 5$ folds of cross-validation (thus corresponding to $10 \times 5 = 50$ GMMs). It may be seen that $-\ln \mathcal{L}_{\text{norm}} \approx 7.6$ for the final best-fitting GMM corresponds to the mode of this distribution, confirming the goodness of fit of the final GMM and its suitability for further use.

### 6.4 Results

This section presents the results of applying the methodology described in the previous section to the high-bandwidth datasets described in Section 6.2. We compare the
following methods for setting the novelty threshold:

- The conventional method of thresholding the pdf, \( p(x) \) (described in Section 3.4).
A threshold is set \( \kappa = -\ln p(x) \).

- The new multimodal, multivariate EVT-based method described in Chapter 5. A
threshold is set \( \kappa = -\ln[1 - P_e(x)] \), where \( P_e(x) \) is the cdf associated with the
estimated EVD.

As described in Section 2.7, the training/validation datasets (\( A_1 \) and \( A_2 \)) are used to
determine the values of metaparameters. In this case, the value \( \kappa \) is the metaparameter,
for each of the three approaches listed above. With the value \( \kappa \) fixed, the methods are
then applied to test data (\( A_3 \) and \( A_4 \)) to determine their performance on previously
unseen data.

We note that, due to the limited amount of data available for training, validation,
and testing, we optimise the value of the metaparameter \( \kappa \) over all speed-bands, and
all shafts. This assumes that the data distributions are comparable within each speed-
band, and that we assume a similar level of “significance” to be abnormal in each, such
that similar novelty thresholds may be used in each speed-band. A possible extension
to this procedure, which could be performed if more data were available, is to opti-
mise the values of the metaparameters for each speed-band and for each engine shaft
independently.

6.4.1 Training and validation

Dataset \( A_1 \)

Figure 6.13 shows NeuroScale projections of data in speed-band \( S_2 \) from each run of
dataset \( A_1 \). The centres of the kernels in the GMM after each run are shown projected
in the plots using red crosses. After runs \( A_{1,1} \) and \( A_{1,2} \), cross-validation selected \( K = 3 \)
kernels as providing the lowest overall cross-validation error. After runs \( A_{1,3} \) and \( A_{1,4} \),
\( K = 4 \) kernel centres were selected. Runs \( A_{1,5} \) and \( A_{1,6} \) were deemed “abnormal”, and
so the GMM was not retrained after these runs. Thus, the projected kernel centres in
the subplots for these two runs are those retained from run \( A_{1,4} \). Run \( A_{1,7} \) was deemed
“normal”, resulting in retraining of the GMM, for which \( K = 5 \) kernels were selected.
Figure 6.13: Visualisation of data from speed band $S_2$ of dataset $A_1$, with runs $A_{1,1}, \ldots, A_{1,8}$ shown in (a) to (h), respectively. Data from each run are shown in black; GMM model centres determined after using 5-fold cross-validation after each “normal” test are shown by red crosses. The model of normality was not updated after runs 5, 6, and 8.
Finally, run $A_{1,8}$ was deemed “abnormal”, and so the GMM was not retrained after this run.

The NeuroScale projections for runs $A_{1,5}$, $A_{1,6}$, and $A_{1,8}$ show significant deviation from the locus of projected data in earlier “normal” runs, corresponding to the relative abnormality of data observed during runs $A_{1,5}$, $A_{1,6}$, and $A_{1,8}$.

In order to examine the behaviour of the EVT-based thresholds, we will now consider example data from a number of runs in dataset $A_1$ in more detail. For the purposes of illustration, plots in the remainder of this section show example periods of engine operation with example novelty thresholds. These example thresholds were set using values of $\kappa$ (and $m$ for the EVT method) which will later be shown to be optimal, based on the available training and validation data.

Novelty scores from a period of “normal” engine operation (from run $A_{1,2}$) are presented in Figure 6.14, where the engine is operating within speed-band $S_1$ during the interval shown. Novelty scores $-\ln p(x)$ based on the probability density values $p(x)$ of the data at indices 39400 and 40500 take values close to $-\ln p(x) = 30$ (see centre plot), which approaches the example novelty threshold for the conventional method, set at $-\ln p(x) = \kappa = 35$. The novelty scores assigned using the new EVT-based method are shown in the lower plot, with the example threshold set at $-\ln[1 - P_e(x)] = \kappa = 12$.

Novelty scores from a period of “abnormal” engine operation during run $A_{1,5}$ are shown in Figure 6.15. Throughout this interval, the engine is operating in speed-band $S_4$, as shown in the upper plot. Novelty scores for periods that engine experts labelled as corresponding to engine abnormality are shown in red. Again, a novelty threshold for the conventional method is shown at $\kappa = 35$ (the same as shown in the previous example, noting that we optimise the metaparameters over all speed-bands and shafts, as described at the beginning of this section). While many of the data points in the “abnormal” interval fall above the novelty threshold, many fall below it. The lower plot shows novelty scores for the EVT-based method, which shows much better separation between “normal” and “abnormal” periods. Note that some “abnormal” data still fall below the novelty threshold near the end of the interval shown; however, no “normal” data exceed the novelty threshold.

Figure 6.15 shows the windowing effect of using the EVT-based approach, in which
the extremum of a window of data persists for $m$ data points. (The selection of a suitable value of $m$ is considered later in this section.) The figure also shows that “extreme” data tend to take much higher novelty scores using the conventional pdf-based method than with the EVT-based method. This is because it is comparably easy for data to take low probability densities $p(x)$ (and hence have high novelty scores) with the conventional method, because the functional form of the GMM is a negative-exponential (3.26) which rapidly approaches zero away from the modes of the distribution. It is comparatively more difficult for data to take low values of probability with respect to the extreme value distribution $P_e(x)$, because the EVD has the majority of its probability mass (its “support”) in the tails of the GMM (see, for example, Figure 5.13), and hence novelty scores for the EVT-based method are generally lower than those for the conventional method.

After showing evidence of “abnormal” behaviour in runs $A_{1.5}$ and $A_{1.6}$, the engine operates normally throughout all of run $A_{1.7}$. Figure 6.16 shows data from the end of this run. As may be seen in the upper plot, the engine starts operating in speed-band $S_4$, then decelerates into speed-band $S_3$, where it remains for the majority of the interval shown, before finally decelerating through speed-bands $S_2$ and $S_1$. The centre plot shows novelty scores with a threshold again set at $\kappa = 35$ for the conventional method. It may be seen that data points from the end of the interval, during which the engine is decelerating, exceed the novelty threshold. By comparison, the threshold based on the EVT-based method, shown in the lower plot and again set at $\kappa = 12$, is not exceeded by these data points during this “normal” period of operation.

Finally, experts deemed that the data show evidence of a significant engine event during run $A_{1.8}$, as shown in Figure 6.17. It may be seen from the upper plot that the engine was rapidly accelerated and decelerated at the end of the period shown. Novelty scores and a threshold for the conventional method, again set at $\kappa = 35$, are shown in the centre plot. The lower plot shows the novelty threshold again set at $\kappa = 12$ using the EVT-based method, where the separation between “normal” and “abnormal” data is greatly improved compared with that shown for the conventional method.
Figure 6.14: Results from an episode during run $A_{1,2}$ showing “normal” data. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 

(a) $\omega_{IP}$ from an episode of run $A_{1,2}$

(b) $p(x)$ for conventional method

(c) $P_e(x)$ for EVT-based method
Figure 6.15: Results from an episode during run $A_{1.5}$ showing “normal” data and “abnormal” in black and red, respectively. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 

113
Figure 6.16: Results from an episode during run $A_{1.7}$ showing “normal” data. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 
Figure 6.17: Results from an episode during run $A_{1.8}$ showing “normal” data and “abnormal” in black and red, respectively. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 
Figure 6.18: Visualisation of data from speed band $S_4$ for dataset $A_2$. Runs $A_{2,1}$, $A_{2,6}$, $A_{2,7}$, and $A_{2,11}$ are shown in subplots (a) to (d), respectively.

Dataset $A_2$

NeuroScale projections for data in speed-band $S_2$ for a selection of runs from dataset $A_2$ are shown in Figure 6.18. Data from the first run $A_{2,1}$ are projected into a cluster centred on the origin. The projections for several subsequent runs remain approximately similar to this, and data from run $A_{2,6}$ are shown projected in a similar cluster. Evidence for the occurrence of precursor events was deemed by engine experts to be visible in the next run, and the projection in the figure for $A_{2,7}$ shows that its constituent data no longer form a tight cluster at the origin, and are instead shown projected in the upper-right of the plot. The final engine event in run $A_{2,11}$ causes a similar deviation, as is shown in the figure.

We now briefly consider two periods from this dataset in more detail. Figure 6.19 shows a period of “normal” operation during run $A_{2,2}$, in which the engine speed falls within speed-band $S_4$. The centre plot shows novelty scores $-\ln p(x)$ and a novelty threshold for the conventional method again set at $\kappa = 35$. It may be seen that the level of the novelty scores for “normal” data is close to the threshold, and temporarily
exceed it at data index 1750. The novelty scores for the EVT-based method, shown in the lower plot, are consistently low throughout the interval, and do not approach the novelty threshold set at $\kappa = 12$. The difference between the two indicates that data can take low probability densities, and hence high novelty scores $-\ln p(x)$, even though they are considered “normal” with respect to the EVD, as was previously seen for the examples shown from dataset $\mathcal{A}_1$.

An example of “abnormal” engine behaviour for this engine is shown in Figure 6.20, which shows a period from run $\mathcal{A}_{2,10}$ that experts classified as corresponding to a precursor to the engine event that occurred in the subsequent run. While “abnormal” data points (shown in red in the figure) exceed the novelty threshold, several data points which were deemed “normal” by engine experts (shown in black), occurring prior to the abnormality, also exceed the novelty threshold. The corresponding plot showing novelty scores and novelty threshold determined using the EVT-based method shows better separation between “normal” and “abnormal” data, such that the example novelty threshold set at $\kappa = 12$ can distinguish between the two types of data. Again, the windowing effect introduced by the use of the EVT-method, in which the extremum in a window persists for $m$ data points, can be seen in the figure.
Figure 6.19: Results from an episode during run $A_{2,2}$ showing “normal” data. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 

\[ -\ln p(x) \] 

\[ -\ln [1 - P_e(x)] \]
Figure 6.20: Results from an episode during run $A_{2,10}$ showing “normal” data and “abnormal” in black and red, respectively. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 
Determining metaparameter values

In order to determine suitable values of the metaparameter $\kappa$ for both of the threshold-setting methods, we consider the performance of each method when classifying “normal” and “abnormal” data over a range of novelty thresholds. We consider the following four types of data:

- **Type $T_1$: training data** - all data in the training sets from which GMMs were constructed (i.e., from “normal” runs), as described previously.

- **Type $T_2$: speed-rejected data** - all data that were rejected by the speed-based rejection algorithm (from “normal” runs), described in Section 6.3.4.

- **Type $T_3$: “normal” validation data** - all periods of data from tested runs that were considered “normal” by engine experts.

- **Type $T_4$: “abnormal” validation data** - all periods of data from tested runs that were considered “abnormal” by engine experts.

Data types $T_1$, $T_2$, and $T_3$ correspond to “normal” data, which should ideally be placed below a novelty threshold, while data type $T_4$ corresponds to “abnormal” data, which should ideally be placed above a novelty threshold. As described earlier in this section, the parameters of the GMM for each speed-band, including the number of kernels $K$ in the mixture, are set automatically at the end of each run. We now need to consider how to optimise the values of the metaparameters $\kappa$ and, for the EVT-based method, $m$. As described at the beginning of this section, the limited quantity of data available for training, validation, and testing means that we optimise $\kappa$ and $m$ over all speed-bands and shafts. If a larger quantity of data were available, it may be possible to perform optimisation of the metaparameters for each speed-band and engine shaft separately$^4$.

Figure 6.21(a) shows the proportion of data for each of these types that exceeded novelty thresholds set using the conventional thresholding method, for increasing values of $\kappa$. It may be seen that this method has great difficulty in separating “normal”

$^4$We note that the maximum value of $\kappa$ for the EVT-based method that can be considered is $\kappa \approx 35$ due to the precision of the computing resources available. It is coincidental that this value is similar to the $\kappa = 35$ optimal value for the conventional pdf-based method.
validation data ($T_3$) from “abnormal” validation data ($T_4$). Indeed, for $\kappa < 16$ there is a greater proportion of $T_3$ data above the novelty threshold than $T_4$ data. For $\kappa > 16$, the method begins to correctly classify more $T_3$ data, at the expense of correctly classifying an ever-decreasing proportion of $T_4$ data. It may be seen that $T_1$ and $T_2$ data (“normal” data from training, and speed-rejection, respectively) are mostly correctly classified for $\kappa > 20$. Noting that, as described in Chapter 2, the priority for an engine health monitoring is a low false-alarm rate in order to avoid grounding a flight unnecessarily, we select a threshold value of $\kappa = 35$ for the conventional method, which gives the maximum possible classification of “abnormal” data while holding the number of incorrectly-classified “normal” data to a minimum.

Figure 6.21(b) shows classification results for the EVT-based method with parameter $m = 5$. This method is much better at separating “normal” from “abnormal” data than the conventional density-thresholding method. The number of $T_3$ data above the novelty threshold is lower than the number of $T_4$ data above the novelty threshold for all values of $\kappa$. However, the misclassification rate of $T_3$ data remains high throughout the range of $\kappa$. The proportion of $T_1$ and $T_2$ data above the novelty threshold is low for $\kappa > 15$.

Classification results for the EVT-based method with parameter $m = 30$ are shown in Figure 6.21(c), which show much better separation of “normal” and “abnormal” data than the previous two plots. For a novelty threshold set at $\kappa = 12$, less than 0.01% of all $T_3$ data are misclassified, while 86% of $T_4$ data are correctly classified. The proportion of $T_1$ and $T_2$ data above the novelty threshold is low for $\kappa > 7$.

Figure 6.21(d) shows results for the EVT-based method when $m = 100$, which shows much more rapid decrease in the number of $T_4$ data correctly classified above the threshold with increasing $\kappa$.

This coarse search over the range of values of $m$ indicates that the value of $m = 30$ provides acceptable results, which is similar to the approach of Hann (2008) in which the value of parameters used for density estimation in multivariate vital-sign data were determined.

We note that recent work in determining the analytical form of the EVDs for multivariate models (Clifton et al. 2009, Hugueny et al. 2009) may make it possible to perform full searches over a range of values of $m$ in a rapid manner.
Thus, we select $\kappa = 12$ and $m = 30$ for the EVT-based method as being those values that give the best classification of “abnormal” data while minimising the number of “normal” data incorrectly classified.
Figure 6.21: Classification results using the conventional and EVT-based methods for datasets $A_1 \ldots A_2$, over a range of values of threshold parameter $\kappa$, showing the proportion of data from four classes that fall above the novelty threshold. The four classes are (i) “normal” data used for training, (ii) “normal” data discarded by speed-based rejection during pre-processing, (iii) “normal” validation data, and (iv) “abnormal” validation data, which are shown by $\times$, $+$, $\cdot$, and a black line, respectively.
6.4.2 Testing with fixed hyperparameters

We now show results of novelty detection using the values of metaparameters fixed by the training/validation process.

Dataset $\mathcal{A}_3$

A precursor event for this engine was deemed to have occurred during the penultimate run, $\mathcal{A}_{3,6}$, a period of which is shown in Figure 6.22. During the period shown, the engine is decelerating from speed-band $S_2$ to $S_1$. The centre and lower plots show that the “abnormal” data corresponding to the precursor event are detected by both the conventional and the EVT-based methods.

The proportion of each type of data that fell above the novelty thresholds for each method is shown in Table 6.3. Both methods correctly classify the “normal” training and speed-rejected data ($T_1$ and $T_2$). While both separate the $T_3$ and $T_4$ data quite well, the EVT-based method has the lowest proportion of “normal” test data ($T_3$) above the novelty threshold, and the highest proportion of “abnormal” test data ($T_4$) above the novelty threshold.

Dataset $\mathcal{A}_4$

Figure 6.23 shows an example from the last “normal” run of this engine, $\mathcal{A}_{4,4}$, where it may be seen that the engine speeds pass through all four speed-bands, $S_1$ to $S_4$. The centre plot shows that the novelty threshold set using the conventional method is exceeded by some of the “normal” data, which are correctly placed below the novelty threshold by the EVT-based method. One of the precursor events identified by engine experts, occurring in run $\mathcal{A}_{4,6}$, is shown in Figure 6.24, which it may be seen that both the conventional and the EVT-based methods detect.

Table 6.3: Classification results for novelty detection methods with test dataset $\mathcal{A}_3$, using metaparameter values set during training/validation. Proportion of each data type above novelty threshold is shown.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Conventional</th>
<th>EVT-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>$&lt; 10^{-3}$</td>
<td>$&lt; 10^{-3}$</td>
</tr>
<tr>
<td>$T_2$</td>
<td>$&lt; 10^{-3}$</td>
<td>$&lt; 10^{-3}$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>0.011</td>
<td>0.002</td>
</tr>
<tr>
<td>$T_4$</td>
<td>0.721</td>
<td>0.922</td>
</tr>
</tbody>
</table>
Figure 6.22: Results from an episode during run $A_{3.6}$ showing “normal” data and “abnormal” in black and red, respectively. (a) Speeds $\omega_{IP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 
Figure 6.23: Results from an episode during run $A_{4,4}$ showing “normal” data. (a) Speeds $\omega_{LP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 
Figure 6.24: Results from an episode during run $A_{4,6}$ showing “normal” data and “abnormal” in black and red, respectively. (a) Speeds $\omega_{LP}$, showing speed-bands as the shaded background; (b) $p(x)$ used with the conventional method, and a novelty threshold (shown by the horizontal black line) set at $\kappa = 35$; (c) $P_e(x)$ and novelty threshold (horizontal black line) determined using the EVT-based method, with a threshold set at $\kappa = 12$. 

127
Table 6.4: Classification results for each multivariate novelty detection method for test dataset $A_4$, using metaparameter values set during training/validation. Proportion of each data type above novelty threshold is shown.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Conventional</th>
<th>EVT-based method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>$&lt; 10^{-3}$</td>
<td>$&lt; 10^{-3}$</td>
</tr>
<tr>
<td>$T_2$</td>
<td>$&lt; 10^{-3}$</td>
<td>$&lt; 10^{-3}$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>0.039</td>
<td>0.004</td>
</tr>
<tr>
<td>$T_4$</td>
<td>0.400</td>
<td>0.870</td>
</tr>
</tbody>
</table>

Overall classification results for this dataset are shown in Table 6.4. Compared with the results shown for dataset $A_3$, it may be seen that both methods misclassify more “normal” $T_3$ and more “abnormal” $T_4$ data. Again, while both methods correctly classify most of the training and speed-rejected “normal” data ($T_1$ and $T_2$), the EVT-based method has the lowest overall misclassification rate.

### 6.5 Conclusion

We have shown that, for the datasets considered in this chapter, novelty detection using the EVT-based method proposed in Chapter 5 can better separate “normal” from “abnormal” data than existing conventional methods, in which a threshold is placed on the model’s pdf, $p(x)$.

Using the limited quantity of data available for this investigation, we have identified values of metaparameters for the proposed EVT-based method that minimise misclassification rates of “normal” and “abnormal” data, as labelled by engine experts. We have shown how model parameters, which are engine specific, such as the number of GMM kernels $K$, can be determined automatically after each engine run. We have shown how the value of the metaparameters $\kappa$ and, for the EVT method, $m$ may be set by considering novelty detection performance over a range of possible values of the metaparameters.

The method based on the multivariate, multimodal extension to EVT (described in Chapter 5) effectively performs windowing of the data, because an “extreme” data point dominates the novelty score for $m$ data points, as shown in Chapter 4. However, unlike heuristic windowing methods (Tarassenko et al. 2005, Tarassenko et al. 2006, Hann 2008), the EVT-based approach offers a probabilistic interpretation because the
probability that the data occurring in a window will exceed a certain novelty threshold is explicitly determined. Varying the value of \( m \) changes the novelty detection performance of the system. As shown in Chapter 4, setting \( m = 1 \) is equivalent to thresholding the pdf \( p(x) \), as is performed with the conventional method. The value of \( m \) should be set according to the dynamics of the system being monitored, which may be performed using data-driven approaches as described in this chapter.

A second effect contributes towards the improved ability of the EVT-based method to discriminate between “normal” and “abnormal” data when compared with the conventional method. As shown in Chapter 4, as \( m \) is increased, the probability mass of the EVD \( p_e(x) \) is placed further from the modes of the generative distribution \( p(x) \). As shown by some of the examples provided in this chapter (such as that shown in Figure 6.19), it is possible for “normal” data to take relatively high novelty scores using the conventional method, because the pdf \( p(x) \) rapidly tends to zero, while taking relatively low scores in the EVT-based method, as discussed in Section 5.4. This provides the EVT-based method with increased robustness to “normal” variability in data, and thus results in improved novelty detection performance.

Finally, we note that it is difficult to select a value of the metaparameter \( \kappa \) that performs well for different engines using the conventional pdf-thresholding method because the contours of the pdf \( p(x) \) are highly data-dependent, and may change from engine to engine. This lack of a suitable value of the metaparameter for the conventional method makes it difficult to set novelty thresholds automatically for monitoring new engines, as is required for engine-specific novelty detection in practice. In contrast, we have shown that it is possible to determine values of the metaparameters for the EVT-based method that provide good novelty detection performance for different speed-bands and engine shafts, for different engines. This suggests that it is possible to perform engine-specific novelty detection using these values of metaparameters in practice, for previously-unseen engines.
Chapter 7

High-Bandwidth Novelty Detection with Full Spectral Data

7.1 Introduction

Chapters 5 and 6 have described the identification of abnormal events by modelling the distribution of tracked orders and other features extracted from vibration spectra. However, many modes of failure manifest themselves as changes in vibration spectra that are not related to energy within these expected frequency ranges.

An example of this is the failure of engine bearings, which are small ball-bearings enclosed within fixed cages such that they may rotate freely. These are used to form load-bearing contacts between the various rotating engine shafts, and they maintain the position of the shafts relative to one another. Damage to the surfaces of these bearings may result in previously-unobserved vibration energy at high frequencies, significantly removed from the narrow frequency bands of the tracked orders observed under “normal” conditions. Failure of the cages in which bearings are mounted can result in constant peaks in spectral energy at previously-unseen multiples of the fundamental tracked orders (Liu & Iyer 1993, Tandon & Choudhury 1999). The latter could be described as novel tracked orders (NTOs) because they are peaks in vibration energy within narrow frequency bands, and are thus tracked orders, but occur at frequencies for which tracked orders are not observed under “normal” conditions.

This chapter describes a method for identifying NTOs and other abnormal content in spectral data, allowing the identification of modes of failure that methods described in previous chapters cannot detect. Principled methods are used for modelling time-series of spectral data observed under “normal” conditions. The goal is to learn an engine-
Table 7.1: Datasets used in the investigation of novel tracked orders.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Engine</th>
<th>Runs</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>B₁</td>
<td>3-shaft class A, engine I</td>
<td>4</td>
<td>B₁₃, B₁₄ contain bearing-cage failure</td>
</tr>
<tr>
<td>B₂</td>
<td>3-shaft class B, engine I</td>
<td>4</td>
<td>B₂₄ contains bearing failure</td>
</tr>
<tr>
<td>B₃</td>
<td>3-shaft class A, engine II</td>
<td>6</td>
<td>Normal engine data</td>
</tr>
</tbody>
</table>

specific model of normality on-line, to provide sensitive novelty detection without the need for tuning heuristic parameters.

7.2 Exemplar Spectral Data

The investigation described by this chapter uses three datasets of full spectral data, described in Table 7.1. The first two engine runs in dataset B₁ and the first three engine runs in dataset B₂ were deemed to be “normal” by engine experts. Runs B₁₃ and B₁₄ (the final two runs in dataset B₁) were deemed to contain evidence of bearing cage failure, in which the cage surrounding an IP-shaft bearing was damaged. Run B₂₄ (the fourth and final run in dataset B₂) was deemed to contain evidence of bearing failure, in which the surface of a bearing between the LP and IP shafts was heavily damaged. Dataset B₃ contains data recorded from an engine with no bearing failures, which is used to investigate the performance of the method described by this chapter when presented with “normal” data.

Following the train-validate-test methodology described in Chapter 2, we partition these datasets into a training/validation set, a test set containing examples of “abnormal” engine operation, and a test set containing only examples of “normal” engine operation, as follows:

- The training/validation set comprises all runs from dataset B₁, from which suitable values of novelty thresholds will be determined such that the α and β (FP and FN) error rates are minimised.

- The “abnormal” test set comprises all runs from dataset B₂, which will be used to evaluate the novelty thresholds fixed during training/validation. Dataset B₂ contains an example of “abnormal” engine operation which was deemed to be more subtle in its effect on vibration spectra than that contained in dataset B₁.
We use $B_2$ for testing to ensure that the resultant novelty detection system is able to detect such subtle failures.

- The “normal” test set comprises all runs from dataset $B_3$.

Due to the limited number of runs available in these full-spectra datasets, the scheme for constructing models of normality described in Chapter 2 must be modified. As shown in Figure 7.1, for each dataset, data acquired during the first two runs are used to construct an engine-specific model of normality. In each case, data acquired from the remaining runs are compared to the model of normality for that dataset.

### 7.3 Dynamical Noise-Floor Estimation

#### 7.3.1 The need for noise-floor estimation

In order to identify NTOs within time-series of spectra, we must first identify and disregard spectral components corresponding to background noise. In existing heuristic methods, this has been performed by defining a single “noise-floor” threshold on spectral energy, below which spectral components are deemed to be due to background noise. A single value for this threshold has been applied to all gas-turbine engines of a similar class, using expert knowledge (Hayton et al. 2003). In order to avoid large numbers of false-positive classifications during novelty detection when using this single threshold with an entire class of engines, it must be set conservatively due to inter-engine vari-
ability within that class. However, this causes subsequent novelty detection to have low sensitivity, because the conservative threshold results in a high number of false-negative classifications.

Furthermore, as will be shown in this section, a constant noise-floor threshold applied to all frequencies of spectral data results in an inability to determine significant engine events that manifest as changes in spectral energy at frequencies much higher than the first few harmonics of fundamental tracked orders. In order to detect bearing failure and other events occurring at higher frequencies, a dynamic approach to noise-floor estimation is required.

### 7.3.2 Investigating background noise

Previous chapters have shown that the vibration amplitude of tracked orders can be characterised with respect to the operating point of the engine; i.e., the rotational speed of the engine shafts. We here extend this approach to examining whole vibration spectra with respect to engine speed, which we will use both for determining engine-specific noise-floor thresholds, and ultimately to construct a model of normality for use in novelty detection.

Figure 7.2 shows the mean vibration amplitude in each of the $f_i, i = 1 \ldots N_{FFT}$ spectral bins computed using all data from run $B_{1,1}$ (here, $N_{FFT} = 1,024$), collected in 1% sub-ranges of shaft speed $\omega_{LP}$. Energy associated with the fundamental tracked orders appears in the lowest decile of the frequency range, from shaft speeds between 15% and 85% of maximum (noting that the engine is idle when $\omega_{LP} < 30\%$). Energy associated with the second harmonic tracked orders appears in the second decile of the frequency range (though at lower amplitudes than the fundamental tracked orders), while a single high-energy harmonic can be seen to extend between 20% and 48% shaft speed, between spectral bins $f_{180}$ and $f_{230}$. Low-amplitude energy may be seen occurring between 60% and 80% shaft speed, between spectral bins $f_{600}$ and $f_{800}$.

Characterising the spectral energy in each spectral bin across the full range of shaft speeds, as illustrated in Figure 7.2, provides a useful summary of engine vibration energy during the “training period” in which the data used to construct the summary, or model, are acquired. Thus, a representation of this form can be used for novelty detection, because we anticipate that “abnormal” vibration energy observed during testing will
Figure 7.2: A speed-based representation of spectral data, showing the mean vibration amplitude from run $B_{1,1}$ collected in 1% sub-ranges of LP shaft speed. In this figure, the speed-frequency space is quantised into $10 \times 10$ bins, shown in red. The vertical axis shows spectral bin indices $f_i$ ranging between $[0 f_s/2]$, where, for this dataset, $f_s = 6.25$ KHz. In this figure, data appear in some way abnormal with respect to this representation.

### 7.3.3 Determining speed-frequency quantisation

In order to examine the variation of background noise across the speed-frequency space, we partition it into $N_\omega \times N_f$ bins, indexed $b_\omega = 1 \ldots N_\omega$, $b_f = 1 \ldots N_f$ for increasing shaft speed and frequency, respectively. In order to reduce the effect of speed-frequency quantisation, we wish to maximise $N_\omega$ and $N_f$, causing the bins to be as small as possible. However, the number of data falling within each bin will decrease with increasing $N_\omega$ and $N_f$, and we require sufficient data in each bin to characterise the distribution of noise vibration amplitudes.

Figure 7.3 shows the number of data falling within each speed-frequency bin for run $B_{1,1}$, for two values of $N_\omega$ while $N_f = 100$. The figure shows that most data occur at $\omega_{LP} \approx 18\%$ and $\omega_{LP} \approx 90\%$, corresponding to aircraft idling and cruising, respectively. The interval between these two speeds corresponds to the relatively short periods when the aircraft is climbing or landing, and thus there are few data falling in...
Figure 7.3: Number of data $N$ falling within each speed bin for run $B_{1,1}$ using quantisations $N_\omega = 100$ and $N_\omega = 20$, shown in the upper and lower plots, respectively. A horizontal line is shown at $N = 100$.

this interval. When $N_\omega = 100$ (corresponding to bins that cover 1% of the total speed range), $N \leq 10$ data are observed in each bin between $20\% \leq \omega_{LP} \leq 75\%$. The use of a coarser quantisation, $N_\omega = 20$ (corresponding to bins that each cover 5% of the total speed range), results in approximately 100 data for each bin within the 20% to 75% speed range, which is sufficient for characterising noise amplitude across the entire speed range. Similarly, the value $N_f = 100$ was found to provide a suitable compromise between retaining fine quantisation while ensuring sufficient data are observed within each bin.

### 7.3.4 Examining noise amplitude-frequency relationship

Figure 7.4 shows $\log_{10} |x|$ for vibration amplitude $|x|$ against frequency, for shaft speeds $\omega_{LP} = 30\%, 60\%, \text{ and } 90\%$. Tracked order vibration occurs as spikes in amplitude across the frequency axis, which occupy a small fraction of the total number of FFT components across the frequency range. It may be seen from the figure that no simple relationship exists between $\log_{10} |x|$ and frequency, with the trend of $\log_{10} |x|$ generally increasing for $1 \leq b_f \leq 180$, generally decreasing for $180 \leq b_f \leq 300$, staying approximately constant.
Figure 7.4: $\log_{10} |x|$ for noise amplitudes $|x|$ across the whole FFT frequency range, for $\omega_{LP} = 30\%$, $60\%$, and $90\%$ in the upper, central, and lower plots, respectively. Mean vibration amplitudes $\mu$ are plotted across the frequency range, shown in black. Vibration amplitudes $\mu \pm \sigma$ are shown in grey.
for $300 \leq b_f \leq 600$, and then generally decreasing for $600 \leq b_f \leq 1024$.

The lack of an explicit amplitude-frequency relationship motivates the use of data modelling techniques to estimate this relationship, given a training set of vibration data.

Vibration amplitudes from FFT components falling within three selected speed-frequency bins using the quantisation ($N_\omega = 20$, $N_f = 100$) are shown in Figure 7.5.

Bin $(b_\omega = 16, b_f = 5)$, shown in Figure 7.5(a), covers the speed range $75\% \leq \omega_{\text{LP}} \leq 80\%$. This bin contains the highest vibration energy of all those shown, due to the fundamental tracked orders that fall within it, which peak in amplitude in that part of the speed range covered by the bin. The figure shows that the distribution of vibration amplitudes within this bin is closely approximated by a gamma distribution. Figure 7.6 shows the same distribution over a larger range of vibration amplitudes on the horizontal axis, where the vertical axis is shown on a log$_{10}$ scale. It may be seen from this figure that the number of vibration amplitudes corresponding to the tracked orders (which have vibration amplitudes $0.05 \leq |x| \leq 0.13$ in s$^{-1}$, for the engine from dataset $\mathcal{B}_1$) is small compared with the much larger number of spectral bins containing background noise. Hence, the tracked orders are not visible in Figure 7.5.

Bin $(b_\omega = 16, b_f = 50)$, shown in Figure 7.5(b), covers the same range of shaft speeds as the bin described above, but corresponds to a higher range of frequencies. The figure shows that the distribution of vibration amplitudes within this bin remains closely approximated by a gamma distribution, but is reduced in location and scale$^1$ compared with that of the lower-frequency bin.

Bin $(b_\omega = 18, b_f = 100)$, shown in Figure 7.5(c), again covers the same range of shaft speeds, but corresponds to the highest frequencies in the FFT. The vibration amplitudes contained within this have a gamma distribution, but further reduced in location and scale compared with the lower-frequency bins.

The number of spectral data observed within each of the $20 \times 100$ bins throughout example run $\mathcal{B}_{1,1}$ is shown in Figure 7.7. It may be seen that the number of data in bins of the same shaft speed is similar across the whole frequency range (vertical axis). This is because each observation consists of a spectrum of $N_{\text{FFT}}$ components, and thus each bin is populated by $N_{\text{FFT}}/100$ data for each observation. The figure shows that no

$^1$Where the location and scale of the gamma distribution $\text{Gam}(x|a,b)$ are $a$ and $b$, respectively.
Figure 7.5: Histograms of vibration amplitudes falling within selected speed-frequency bins for run $B_{1,1}$. Histograms are normalised such that they integrate to unity and are approximately pdfs, $p(x)$. ML gamma distributions for each bin are shown in black.

Figure 7.6: Histogram of vibration amplitudes falling within speed-frequency bin $b_\omega = 16, b_f = 5$ for run $B_{1,1}$, corresponding to that shown in Figure 7.5(a) over a greater range of vibration amplitudes on the horizontal axis, to show the presence of tracked order values. These are few in number compared with the number of values corresponding to noise (note the log scale on the vertical axis).
Figure 7.7: Number of data within each of 20 × 1024 speed-frequency bins for run $B_{1,1}$. Noting that most bins contain $N > 100$ data, we assume that the representation is suitable for modelling the distribution of noise within each bin from $15\% \leq \omega_{LP} \leq 90\%$.

### 7.3.5 Modelling noise with gamma distributions

Figure 7.5 illustrates that the vibration amplitude of noise generally decreases with increasing frequency, which is the case for all 20 columns of bins in the $20 \times 100$ representation. Zhang (2007) observed similar behaviour for background noise in acoustic signals, where narrow frequency bands used to transmit speech were seen to contain gamma-distributed background noise.

In order to model the distribution of amplitudes $X_{\omega,f} = \{x_1, \ldots, x_N\}$ within each of the $20 \times 100$ bins considered above, we find the ML gamma distribution, $\text{Gam}_{ML}(x|\hat{a}, \hat{b})$ for which the likelihood $L$ may be written

$$L(a, b) = p(x_1, x_2, \ldots, x_N|a, b)$$

$$= \prod_{i=1}^{N} p(x_i|a, b)$$

$$= \prod_{i=1}^{N} \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-x/b) \quad (7.1)$$

and for which the log likelihood is

$$\ln L(a, b) = Na \ln b - N \ln \Gamma(a) + (a - 1) \sum_{i=1}^{N} \ln(x_i) - b \sum_{i=1}^{N} x_i \quad (7.2)$$
Figure 7.8: Fitting gamma distributions to vibration amplitudes from an example run falling within bin $b_\omega = 16$, $b_f = 5$, which contains energy associated with tracked order vibration. (a) An initial ML gamma distribution fit to all data $X_{16,5}$ within the bin. (b) ML gamma distribution fit to all data $X'_{16,5}$ falling within $\mu \pm 3\sigma$.

Thus, the maximum likelihood parameters $(\hat{a}, \hat{b})$ are given by $\arg\max_{a,b} \ln L(a, b)$. Solutions for this maximisation do not exist in closed form, and we use the iterative methods described in Choi & Wette (1969), where we select initial estimates of the parameters, $a_0$ and $b_0$, using the method of moments,

$$a_0 = \bar{x}^2 / s^2$$  \hspace{1cm} (7.3)

$$b_0 = s^2 / \bar{x}$$  \hspace{1cm} (7.4)

for sample mean and variance $\bar{x}$ and $s$, respectively.

The above method is suited to those bins within the $20 \times 100$ representation that contain only background noise, but its application to data from bins that contain significant vibration energy due to tracked orders (such as bin $b_\omega = 16$, $b_f = 5$, shown in Figures 7.5 and 7.6) may result in a ML gamma distribution that is a poor fit to the noise, as shown in Figure 7.8(a). Note here that the axes have been scaled differently to those shown in Figure 7.5(a) in order to illustrate the gamma distribution for that bin more clearly.

Though the number of data corresponding to tracked order vibration is small compared with the number of data corresponding to background noise (and which, as with Figure 7.5, occur an order of magnitude higher on the horizontal axis than the mode of the gamma distribution), they are sufficiently large in magnitude to cause the fit of the gamma distribution to the background noise to be poor. In order to avoid this, we perform the following operations for each bin:
Figure 7.9: ML gamma mean $\mu$ and standard deviation $\sigma$ for distributions fitted to vibration amplitudes in $b_\omega = 1 \ldots 20$, $b_f = 1 \ldots 100$ speed-frequency bins for run $B_{1,1}$.

1. Fit an initial gamma distribution to all vibration amplitudes $X_{\omega,f}$ within the bin.

2. Determine the mean $\mu$ and variance $\sigma^2$ of this initial gamma distribution,

$$\mu = ab \quad (7.5)$$

$$\sigma^2 = ab^2 \quad (7.6)$$

3. Fit a second gamma distribution to the “trimmed” dataset

$$X'_{\omega,f} = \{ x \mid \mu - 3\sigma \leq x \leq \mu + 3\sigma \} \quad (7.7)$$

The resultant distribution for bin ($b_\omega = 16$, $b_f = 5$) is shown in Figure 7.8(b). After disregarding data lying further than $3\sigma$ from the mean, which includes the vibration amplitudes corresponding to the tracked orders, the fit of the gamma distribution to the noise is much closer.

Figure 7.9 shows ($\mu, \sigma$) from ML gamma distributions fitted using the above method to spectral data from an example run for all $20 \times 100$ bins. It may be seen that the mean and standard deviation generally increase with increasing speed, and generally decrease with increasing frequency, as noted previously for the example bins shown in Figure 7.5.

### 7.3.6 Determining noise-floor thresholds

Using the noise distributions, a threshold $x^+_{\omega,f}$ may now be defined for each of the $20 \times 100$ bins, below which vibration amplitudes will be deemed to be background noise. Thus,
we wish to determine the upper limit of extrema generated from the gamma distribution in each bin. For the univariate gamma distribution \( p(x|a,b) \), the EVD tends towards a Gumbel distribution \( p_e(x|c,d) \). Classical univariate EVT (Embrechts et al. 2008) gives estimates of the Gumbel parameters \((c,d)\) in terms of the gamma parameters \((a,b)\) and \(m\), the number of data drawn from \( p(x|a,b) \),

\[
c = \frac{1}{b} \tag{7.8}
\]

\[
d = \frac{1}{b} \left( \ln m + (a - 1) \ln \ln m - \ln \Gamma(a) \right) \tag{7.9}
\]

Given a spectrum of \( N_{\text{FFT}} \) components for each observation, each bin will contain \( N_{\text{FFT}}/100 \) data. We wish to determine the expected extremum for this number of data points generated within each bin, and thus define \( m = \lceil N_{\text{FFT}}/100 \rceil \).

Figure 7.10 shows data from bin \((b_\omega = 16, b_f = 5)\) for run \(B_{1,1} \), with the corresponding ML gamma distribution \( p(x|a,b) \). The Gumbel EVD \( p_e(x|c,d) \) estimated using (7.8) and (7.9) is also shown, together with the corresponding cumulative Gumbel EVD \( P_e(x|c,d) \), which will be used to set an upper threshold on the noise distribution. This EVT threshold will be the noise-floor estimate for the bin, and thus will vary according to the distribution of noise within each bin.

Figure 7.11 compares the resultant cumulative EVD \( P_e(x|c,d) \) obtained using classical EVT to that obtained using the ML method described in Algorithm 5.1, again using bin \((b_\omega = 16, b_f = 5)\). The ML Gumbel distribution was fitted to \( N = 10^6 \) extrema obtained via sampling using Algorithm 5.2. It may be seen from the figure
that the ML Gumbel distribution closely fits the empirical EVD $\tilde{P}_e(x)$ of the extrema, while the classical EVD slightly underestimates the cdf around the mode of the Gumbel (corresponding to the value of $x$ where $P_e(x)$ has maximum gradient). However, we are interested in setting a threshold close to $P_e(x) = 1$ (e.g., $P_e(x) = 0.99$), where the classical and ML EVDs are both close to the empirical EVD, and thus both approaches are suitable for this task.

Thresholds set at $P_e(x) = 0.99$ using both classical and ML EVT methods are shown in Figure 7.12, expressed in terms of vibration amplitude, where it may be seen that both methods give similar thresholds. Thresholds obtained with ML EVT, shown in
Figure 7.13: Vibration amplitudes and EVT noise-floor thresholds (shown in red) for a bin containing significant energy associated with tracked orders (“TO bin” $b_\omega = 8, b_f = 5$) and a bin containing only noise (“non-TO bin”, $b_\omega = 8, b_f = 40$). (a) and (b) show histograms of vibration amplitudes for the TO and non-TO bins, respectively, plotted on linear axes. (c) and (d) show the same using $\log_{10}$ vertical axes.

Figure 7.12(b), are slightly higher than those obtained with classical EVT, shown in Figure 7.12(a). Due to its much simpler computation, we choose to set thresholds $x_{\omega,f}$ in each of the $20 \times 100$ windows using the classical EVD.

We now examine the effect of the EVT noise-floor threshold on bins containing vibration energy associated with tracked orders. Figure 7.13(a) shows a histogram of vibration amplitudes and the noise-floor threshold for a bin containing tracked-order energy, determined using classical EVT. Figure 7.13(b) shows the same for a bin containing no tracked-order energy. Note that the horizontal axes for Figure 7.13(a) and (b) differ by an order of magnitude. It may be seen that the noise-floor threshold occurs at $x_{8,5}^+ = 0.014$ in s$^{-1}$ and $x_{8,40}^+ = 0.0033$ in s$^{-1}$ for the former and latter, respectively. Tracked order vibration amplitudes result in a “heavy tail” for the distribution of these amplitudes in the former.

Plotting the histograms on a logarithmic scale shows that the tail of the distri-
bution of vibration amplitudes for the bin containing tracked order energy, shown in Figure 7.13(c), extends above the noise-floor threshold, while the distribution for the window containing no tracked order energy, shown in Figure 7.13(d), does not and will be deemed to be entirely background noise.

### 7.4 Testing Spectral Data

This section considers how spectral data may be tested with respect to a model of normality. Dataset $\mathcal{B}_1$ is used throughout this section.

Consider now a test sample which consists of $X = \{x_1, \ldots, x_{N_{\text{FFT}}}\}$ spectral components, observed at LP shaft speed $\omega_{\text{LP}}$.

In order to evaluate this sample with respect to the noise-floor thresholds determined using the training dataset, we find the column of bins with speed index $b_\omega$ given by

$$b_\omega = \left\lceil \frac{\omega_{\text{LP}} - \omega_{\text{min}}}{W_\omega} \right\rceil$$

where $W_\omega$ is the range of shaft speeds covered by each window, given by

$$W_\omega = \frac{\omega_{\text{max}} - \omega_{\text{min}}}{N_\omega}$$

and where $\omega_{\text{min}}$ and $\omega_{\text{max}}$ are the minimum and maximum LP shaft speeds over which the $N_\omega$ bins span, respectively. For the investigation described by this chapter, $\omega_{\text{min}} = 0\%$, $\omega_{\text{max}} = 100\%$, and $N_\omega = 20$. Similarly, the index of the bin $b_f$ into which each of the $f_i$, $i = 1 \ldots N_{\text{FFT}}$ spectral components falls is given by

$$b_f = \left\lceil \frac{f_i}{W_f} \right\rceil$$

where $W_f$ is the extent of the frequency range (in spectral components) covered by each bin, given by

$$W_f = \frac{N_{\text{FFT}}}{N_f}$$

for the $N_f = 100$ bins.

Thus, each spectral component $x_i$ in $X$ is evaluated with respect to the EVD of the appropriate bin. Figure 7.14(a) shows spectra of vibration amplitudes for part of run $\mathcal{B}_{1,1}$, in which the engine initially accelerates at a high rate from time index $t = 50$ to $t = 180$, and then more gradually until $t = 700$. Fundamental tracked orders may
Figure 7.14: Evaluating spectra from a test run with respect to noise-floor estimates. (a) Vibration amplitudes for test run $B_{1,1}$, shown as a time-series of vibration spectra. (b) Probabilities $P_e(x)$ for each spectral component evaluated with respect to cumulative EVDs from noise distributions in $20 \times 100$ bins. (c) Novelty scores $z(x)$ as defined in (7.14) for each spectral component.
be seen in the lowest decile of the frequency range, varying in frequency proportional to engine speed. Between time indices \( t = 100 \) and \( t = 150 \), significant vibration energy may be seen to occur in the lower three deciles of the frequency range, associated with harmonic tracked-order vibration. Low-amplitude vibration associated with higher-order harmonic tracked orders may be seen throughout the example, appearing as a series of inclined lines extending across the frequency range. Horizontal bands of low-amplitude vibration noise may be seen extending across the frequency range throughout the example.

Figure 7.14(b) shows the result of determining cumulative EVD probabilities \( P_e(x) \) for each spectral component with respect to the EVDs of the \( 20 \times 100 \) bins considered previously. It may be seen that harmonic and fundamental tracked orders alike have \( P_e(x) \approx 1 \), as vibration amplitudes associated with them are considerably higher than the distributions of background noise used to construct the EVDs in the \( 20 \times 100 \) bins. The horizontal bands of low-amplitude noise occurring across the frequency range generally have probabilities \( P_e(x) \ll 1 \).

For the purposes of visualisation, we define a novelty score

\[
z(x) = -\log_{10} \{1 - P_e(x)\} \tag{7.14}\]

Figure 7.14(c) shows \( z(x) \) for each spectral component in the example. The horizontal bands of low-amplitude vibration noise have been attenuated, while the fundamental and harmonic tracked orders appear clearly. Tracked orders can be seen throughout the frequency range as lines with positive gradient, increasing in frequency proportional to the rotational frequency of the engine, which is increasing during this interval. The 14HP harmonic tracked order is visible at the top of the frequency range, and is labelled in the figure.

The relationship between vibration amplitude, \( P_e(x) \), and \( z(x) \) is illustrated in Figure 7.15, which shows all three quantities for spectral data observed at time \( t = 680 \) from the example run considered previously. It may be seen from the plot of vibration amplitudes that only the fundamental and second-harmonic tracked orders are immediately obvious; the fundamental tracked orders occur below spectral component \( f_{75} \), while the large vibration amplitudes of spectral component \( f_{120} \) is due to 2HP.

Figure 7.15(a) shows the disadvantages of using a single noise-floor threshold for
all spectral data: it must be set higher than the vibration amplitude of background noise occurring in lower parts of the frequency range, where the fundamental tracked orders occur, but vibration amplitudes corresponding to higher-order harmonics, occurring higher in the frequency range, fall beneath this single noise-floor threshold. Thus, existing methods typically analyse only the fundamental and first few harmonic tracked orders (Hayton et al. 2003), while the higher-order tracked orders are not observable due to the single noise-floor threshold, even though they are visible as low-amplitude lines in the original time-series of spectra.

Corresponding plots of $P_e(x)$ and $z(x)$ in Figure 7.15(b) and (c) show that the use of
probabilistic noise-floor thresholds that vary across the frequency and speed range make it possible to identify higher-order harmonic tracked orders throughout the frequency range, as they have \( P_e(x) \) and \( z(x) \) values much greater than those of the surrounding spectral components containing background noise. The \( P_e(x) \) plot shows that harmonics of the HP tracked order are visible throughout the frequency range, appearing as spikes in \( P_e(x) \) equally-spaced along the frequency axis, up to 15HP occurring at spectral component \( f_{950} \).

With the method proposed in this chapter, spectral information content not previously available can be automatically extracted from test spectra. Though the goal of this work is to identify novel tracked orders (i.e., spectral information in test data that was not present in the spectral data used for training), we note that the method can also be used to extract “non-novel” higher-order harmonic tracked orders occurring throughout the frequency range.

## 7.5 Identifying Novel Tracked Orders

Spectral components with vibration amplitudes above the probabilistic noise-floor thresholds \( x^{+}_{\omega,f} \) have been identified, and it remains for them to be separated into those which correspond to spectral content previously observed within the training data, and those which are novel with respect to the training data.

### 7.5.1 Characterising spectral energy of training data

In order to identify novel vibration energy in test spectra, we must record the location of vibration energy observed in the training data. Note that we here disregard all spectral components with vibration amplitudes deemed to be “background noise” (using the method described in Section 7.3), and hereafter only refer to those spectral components with vibration amplitudes above the appropriate noise-floor thresholds.

To record the location of vibration energy from the training data in speed-frequency space, we again partition that space into \( N'_\omega \times N'_f \) bins. We will consider a number of possible quantisations, \( N'_\omega = \{5, 10, 20, 40, 60, 80, 100\} \) and \( N'_f = \{10, 50, 100, 200, 400, 500, 1024\} \), thus corresponding to \( 7 \times 7 = 49 \) different speed-frequency resolutions.

Each of the 49 speed-frequency quantisations \((N'_\omega, N'_f)\) defines a matrix of bins
indexed $b_\omega = 1 \ldots N'_\omega$ along the speed axis, and $b_f = 1 \ldots N'_f$ along the frequency axis. We define counting matrix $C_{i,j}$ such that element $(i, j)$ is set to the number of FFT components in the training set that fall within the corresponding speed-frequency bin $(i, j)$.

Figure 7.16(a) and (b) shows the matrices $C_{i,j}$ for two quantisations (one coarse, and one fine), when using the training set of “normal” run data $\{B_{1,1}, B_{1,2}\}$. The figure shows that both $C$ matrices are zero in the lower and upper parts of the speed range for these runs. Vibration energy corresponding to the fundamental and lower-order harmonic tracked orders results in large numbers of FFT components falling in bins in the lower parts of the frequency range, while higher-order harmonics result in larger counts throughout the higher parts of the frequency range. $C_{10,10}$ is obviously a much coarser quantisation of the number of spectral components containing significant vibration energy than $C_{100,100}$.

### 7.5.2 Investigating the $C$ matrix

The $C$-matrix characterises the distribution of spectral energy throughout the speed-frequency space for the training data. We now define a quantity $N_m$ such that any bin $(i, j)$ with $C_{i,j} \geq N_m$ will be considered to contain “known tracked order” vibration energy. Any FFT component from test data (with vibration amplitude above the appropriate noise-floor threshold) falling into that bin $(i, j)$ will be classified “normal”; i.e., it lies in a part of the speed-frequency space for which a sufficient number of FFT components above the noise-floor threshold were observed in the training data.

Conversely, if an FFT component from test data falls into a bin $(i, j)$ for which $C_{i,j} < N_m$, it will be classified “abnormal” with respect to the training data; i.e., it lies in a part of the speed-frequency space where an insufficient number of FFT components above the noise-floor threshold were observed during training.

For the investigation described by this chapter, we consider a range of values $N_m = \{1, 3, 5, 10, 20, 50, 100\}$, for each of the 49 candidate quantisations described previously. Thus, this results in $49 \times 7 = 343$ different combinations of $N'_\omega$, $N'_f$, and $N_m$.

Again using the training set $\{B_{1,1}, B_{1,2}\}$, Figure 7.17 shows the proportion of speed-frequency bins that are deemed to contain “known tracked orders” in each of the 49 candidate quantisations, for two values of $N_m$; i.e., it shows the proportion of bins $(i, j)$
Figure 7.16: $C$ matrices for two different speed-frequency quantisations. (a) $C_{10,10}$ showing the number of spectral data from the training set falling within $10 \times 10$ speed-frequency bins. (b) $C_{100,100}$ showing the number of spectral data from the training set falling within $100 \times 100$ speed-frequency bins.
for which \( C_{i,j} \geq N_m \), for each quantisation.

For the case when \( N_m = 3 \), shown in Figure 7.17(a), the largest proportion of speed-frequency bins containing more than \( N_m \) training data occurs for the most coarse quantisation, \((N'_ω = 5, N'_f = 10)\). For this quantisation, approximately 0.8 of all bins are deemed to contain “known tracked orders”, and thus resultant novelty detection will be maximally insensitive to novel tracked orders.

As the quantisation becomes less coarse, the proportion of bins for which \( C_{i,j} \) exceeds \( N_m \) must decrease because the same number of training data are distributed between a larger number of bins, and thus bins will tend to have lower counts \( C_{i,j} \). This is seen in Figure 7.17, in which increasing \( N'_ω \) and \( N'_f \) corresponds to lower proportions of bins with \( C_{i,j} \geq N_m \).

Figure 7.17(b) shows the case when \( N_m = 50 \). Compared to the previous case, each bin must contain a larger number of spectral data with vibration amplitude above the corresponding noise-floor threshold in order to be deemed to contain a “known tracked order”. The proportion of bins deemed to contain “known tracked orders” decreases when compared to the case when \( N_m = 3 \), and thus resultant novelty detection is more sensitive to novel tracked orders for increasing \( N_m \).

So, as quantisation becomes less coarse, fewer bins are deemed to contain “known tracked orders”, and thus it is easier for test data to be classified “novel”. This will be particularly important when we examine the performance of the resultant novelty detection system: finer quantisations will tend to result in more “novel” classifications,
as just described, and so the false-positive rate will tend to increase correspondingly. While it is desirable to use finer quantisations to improve sensitivity to truly abnormal spectral content, we must balance this against the increasing false-positive rate and choose a sufficiently coarse quantisation such that the false-positive rate is acceptable.

7.5.3 Using the $C$ matrix for novelty detection

This subsection presents results of novelty detection using the datasets described previously in Section 7.2.

We first examine various candidate quantisations (i.e., ranges of $N'_\omega$ and $N'_f$) and candidate values of parameter $N_m$, for dataset $B_1$, and identify suitable values for these three model parameters based on the results of novelty detection.

We then apply the method to test datasets $B_2$ and $B_3$, evaluating the performance of novelty detection using the values of the three model parameters selected during training/validation.

In each case, engine-specific models are constructed for each dataset, as described in Section 7.2.

Training/validation with dataset $B_1$

As described in Section 7.2, runs $B_{1,1}$ and $B_{1,2}$ of dataset $B_1$ are deemed to be “normal”, while runs $B_{1,3}$ and $B_{1,4}$ are deemed to contain evidence of bearing-cage failure. We consider the range of frequency resolutions $N'_f = \{10, 50, 100, 200, 500\}$, the range of speed resolutions $N'_\omega = \{5, 10, 20, 40, 60, 80, 100\}$, and the range of parameter $N_m = \{1, 3, 5, 10, 20, 50, 100\}$.

Vibration spectra from parts of each run in dataset $B_1$ are shown in Figure 7.18. Runs $B_{1,3}$ and $B_{1,4}$ were deemed to contain evidence of suspected bearing-cage failure by engine experts, which is shown in the figure.

We must now investigate the effect of varying the values of parameters on novelty detection performance. Results of novelty detection are shown in Figure 7.19, for all candidate speed-frequency quantisations, and with an initial choice of $N_m = 3$. It may be seen that for “normal” runs $B_{1,1}$ and $B_{1,2}$ the number of false-positive “novel” classifications is low (and, in the case of run $B_{1,2}$, zero) for speed quantisation $N'_\omega \leq 60$ and frequency quantisation $N'_f \leq 200$. 

153
Figure 7.18: Vibration spectra for episodes from runs from dataset $B_1$. Data from runs $B_{1,1}$ to $B_{1,4}$ are shown in subplots (a) to (d). Regions of the spectral plots deemed to contain evidence of “abnormal” engine behaviour are shown in red, for $B_{1,3}$ and $B_{1,4}$. 
Figure 7.19: Results of novelty detection for runs from dataset $B_1$, the first two of which were used to determine noise-floor thresholds and $C$ matrices. Each plot shows the proportion of spectral data from a run that were classified “abnormal” (i.e., corresponding to novel tracked order vibration) for all quantisations, and with $N_m = 3$.

For runs $B_{1,3}$ and $B_{1,4}$, which show evidence of bearing-cage failure, the most coarse speed-frequency quantisation ($N'_\omega = 5, N'_f = 10$) results in no “novel” classifications, which is a false-negative result. For run $B_{1,3}$, the use of quantisations $N'_\omega = \{5, 10, 20\}$ and $N'_f = \{50, 100\}$ results in correct identification of the spectral content associated with bearing cage-failure, as will be illustrated below. For run $B_{1,4}$, using quantisations $N'_\omega = \{5, 10, 20, 40, 60\}, N'_f = \{50, 100\}$ results in correct identification of novelty, as will be shown. These values of the quantisation parameters are indicated in Figure 7.19(c) and (d) as those contained by the dashed lines. Increasing speed or frequency resolution above these values introduces false-positive “novel” classifications, not associated with the bearing cage-failure.

Figure 7.20 shows the $\alpha$ (FP) and $\beta$ (FN) error rates calculated using (2.5) and (2.6), respectively, for a range of values of $N_m$. The use of $N_m = 1$ produces no “novel” classifications for any runs (which is a false negative result), while values of $N_m$ greater than 3 introduce more false-positive “novel” classifications.

From the results shown in Figure 7.20, we confirm our selection of the value $N_m = 3$ for use in novelty detection. We must now select those values of the quantisation pa-
parameters that, given the results obtained using the limited training data available, compromise best between being able to detect “abnormal” engine behaviour while having a low false-positive rate. Of the candidate values for the quantisation parameters shown above, we select $N'_\omega = 20$ and $N'_f = 50$ in line with current practice by engine experts, where the engine speed-range is typically quantised into 20 equal speed subranges (this will be further considered in Chapter 8).

$P_e(x)$ for “abnormal” spectral data using the selected values $N'_\omega = 20$, $N'_f = 50$, and $N_m = 3$ is shown in Figure 7.21. This shows the high-amplitude vibration energy at frequencies corresponding to bearing-cage failure, which is a series of novel tracked orders occurring between $t = 200$ and $t = 400$ for run $B_{1,3}$, and between $t = 300$ and $t = 500$ for run $B_{1,4}$. This abnormal spectral energy is correctly identified using the selected speed-frequency quantisation and $N_m$ value.

As $N'_\omega$, $N'_f$, and $N_m$ increase, the number of false-positive “novel” classifications increases. Figure 7.22(a) and (b) show $P_e(x)$ for spectral data classified “novel” from runs $B_{1,1}$ and $B_{1,2}$ (when $N'_\omega = 100$, $N'_f = 50$, and $N_m = 100$), respectively. This shows that many normal tracked orders are classified “novel” using this quantisation.

Thus, for this dataset, use of the quantisations $N'_\omega = 20$, $N'_f = 50$ and parameter $N_m = 3$ allows identification of novel tracked orders indicative of abnormal engine condition, with minimal false-positive classification.
Figure 7.21: $P_e(x)$ for episodes from runs containing “abnormal” engine behaviour, when $N_m = 3, N'_\omega = 20$, and $N'_f = 50$. Results for runs $B_{1,3}$ and $B_{1,4}$ are shown in (a) and (b), respectively. $P_e(x)$ is shown by the greyscale, indicated to the right of each subplot.

Figure 7.22: $P_e(x)$ for episodes from runs containing “normal” engine behaviour, when $N_m = 100, N'_\omega = 100$, and $N'_f = 50$. Results for runs $B_{1,1}$ and $B_{1,2}$ are shown in (a) and (b), respectively. $P_e(x)$ is shown by the greyscale, indicated to the right of each subplot.
Table 7.2: Proportion of spectral bins in each run of dataset $B_2$ that were classified “novel”

<table>
<thead>
<tr>
<th>Run</th>
<th>Proportion of bins classified “novel”</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{2,1}$</td>
<td>$&lt; 10^{-6}$</td>
</tr>
<tr>
<td>$B_{2,2}$</td>
<td>$&lt; 10^{-6}$</td>
</tr>
<tr>
<td>$B_{2,3}$</td>
<td>$&lt; 10^{-6}$</td>
</tr>
<tr>
<td>$B_{2,4}$</td>
<td>$0.0025$</td>
</tr>
</tbody>
</table>

Figure 7.23: Vibration spectra for episodes from run $B_{2,4}$, with $P_e(x)$ for “novel” spectral data. (a) Vibration spectra from run $B_{2,4}$. (b) $P_e(x)$ for “novel” spectral data from run $B_{2,4}$ when $N_{\nu} = 20, N_{\nu} = 50, \text{ and } N_m = 3.$

**Testing with dataset $B_2$**

Section 7.2 noted that runs $B_{2,1}$ to $B_{2,3}$ were deemed “normal” by engine experts, while run $B_{2,4}$ contained evidence of damage to a bearing, shown as a novel tracked order in the vibration spectra for that run.

As described in Section 7.2, we define the training set of “normal” data to be $\{B_{2,1}, B_{2,2}\}$, from which noise-floor thresholds and the $C$ matrix are determined, using the values of $N_{\nu} = 20, N_{\nu} = 50, \text{ and } N_m = 3$ fixed during the training/validation phase, described above. We then classify all data from all runs in dataset $B_2$ using the $C$ matrix.

The proportion of spectral data in each run that were classified as “novel” for $B_2$ is shown in Table 7.2. This corresponds to an $\alpha$ (FP) error rate of $1\times10^{-6}$ and a (FN) error rate of $\beta \approx 0$, calculated using (2.5) and (2.6), respectively.

Vibration spectra from part of run $B_{2,4}$ are shown in Figure 7.23(a). $P_e(x)$ for “novel”
Table 7.3: Proportion of spectral bins in each run of dataset $B_3$ that were classified “novel”

<table>
<thead>
<tr>
<th>Run</th>
<th>Proportion of bins classified “novel”</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_{3,1}$</td>
<td>$&lt; 10^{-6}$</td>
</tr>
<tr>
<td>$B_{3,2}$</td>
<td>$&lt; 10^{-6}$</td>
</tr>
<tr>
<td>$B_{3,3}$</td>
<td>0.00010</td>
</tr>
<tr>
<td>$B_{3,4}$</td>
<td>0.00021</td>
</tr>
<tr>
<td>$B_{3,5}$</td>
<td>0.00011</td>
</tr>
<tr>
<td>$B_{3,6}$</td>
<td>0.00005</td>
</tr>
</tbody>
</table>

spectral data when $N'_\omega = 20$, $N'_f = 50$, and $N_m = 3$ is shown in Figure 7.23(b). This shows the novel tracked order corresponding to bearing damage at $f \approx 2,500$ Hz, which is correctly identified using this speed-frequency resolution and $N_m$ value.

**Testing using dataset $B_3$**

We have previously identified candidate quantisation parameters and verified their use with an independent “abnormal” dataset, but now wish to evaluate their performance with entirely “normal” data.

As described in Section 7.2, dataset $B_3$ consists of 6 runs deemed to be “normal” by engine experts. We determine noise-floor thresholds and construct a $C$ matrix using the first two runs from the dataset, $B_{3,1}$ and $B_{3,2}$. Similar quantisations were considered to those used previously.

Table 7.3 shows the results of testing all six runs with respect to the noise-floor thresholds and $C$ matrix constructed using the first two runs. Any “novel” classifications made are false positives. It can be seen from the table that the use of the candidate quantisation values identified previously, $N'_\omega = 20$, $N'_f = 50$, and with $N_m = 3$, result in almost no false-positive classifications. The overall $\alpha$ (FP) error rate, calculated using (2.5), is $1.01 \times 10^{-6}$.

**7.6 Conclusions**

We have shown that a speed-frequency representation is suitable for characterising the distribution of spectral energy for normal engine conditions, and have proposed a method of performing novelty detection using spectral data such that a noise model is learned from a training set, and then “novel” vibration energy with respect to the noise model
in test data can be detected.

This technique is automatic, given suitable values for the speed-frequency quantisation and parameter $N_m$ determined through training/validation, and thus allows engine-specific novelty detection to take place, whereas existing methods are generic to classes of engine.

Results of analysis using two datasets that exhibit abnormal vibration energy show that the approach can be used to learn a model of noise from examples of engine operation, such that novel vibration energy can be identified in test data with a minimum of false-positive classifications. Suggested parameters for the noise model have been identified, and the possible occurrence of false positives has been tested using a dataset containing no engine abnormalities.

We also note that the proposed probabilistic method of estimating varying noise-floor thresholds across the speed-frequency representation allows higher-order harmonic tracked orders to be identified, that otherwise could not be highlighted with existing techniques. Thus, the proposed method could also be used for feature extraction of the amplitudes of these higher-order tracked orders. The investigation described in this chapter has shown that tracked orders up to 15HP can be identified. It is possible that using higher sampling frequencies could allow harmonic tracked orders above 15HP to be extracted.
Chapter 8

Low-Bandwidth Novelty Detection

8.1 Introduction

In Chapter 2, we described the tight constraints on bandwidth when using on-wing engine health monitoring (EHM) systems. A novelty detection system that is to be used as part of an on-wing EHM system needs to operate within such an environment, including the low-bandwidth transmission of its outputs to a ground-based analysis system.

This chapter describes a method of low-bandwidth novelty detection suitable for use in modern EHM systems.

8.2 Using Univariate Vibration Signatures

8.2.1 Characterising normality

In Chapter 2, we introduced the vibration signature as a method of characterising the vibration amplitude of a tracked order throughout an engine run as a function of shaft speed. As the vibration signature is constructed for a specific tracked order, this is in contrast with the multivariate approach described in Chapter 5 in which information from multiple tracked orders was fused together into a single model. Typically, signatures are constructed for the fundamental tracked orders (1LP, 1IP, and 1HP), as these tracked orders are most likely to be indicative of abnormalities in engine operation.

Though vibration signatures of the same tracked order may vary significantly between different engines, due to the variability in behaviour between engines, they are typically similar when constructed using data from consecutive runs of the same engine (Nairac et al. 1999, Clifton 2006), and thus may be used to characterise "normal" engine
Figure 8.1: A signature of the amplitude of the 1LP tracked order $|x|$ against LP shaft speed, $\omega_{LP}$, where the colour scale on the right-hand side of the plot indicates the value of the distribution function $p(|x|)$ in each speed bin. The signature in this example has $N_\omega = 400$ speed bins over the range of LP shaft speeds $[0\ 100]%$ max. Novelty thresholds $x_\kappa$ in each speed bin are shown as a red line, shown here for the purposes of illustration, and the mean vibration amplitude in each speed bin is shown as a yellow line.

condition using engine-specific models.

Figure 8.1 shows an example signature of vibration amplitude for the 1LP tracked order (see Section 2.5) where the $\omega-$axis is quantised into $N_\omega = 400$ speed bins. A novelty threshold $x_\kappa$ may be defined in each of the speed bins using, for example, the methods described in this chapter. These thresholds for each bin thus define an envelope of acceptable “normal” vibration amplitude for that tracked order. New vibration (test) data may then be classified “abnormal” if they exceed these novelty thresholds in any of the $N_\omega$ bins.

### 8.2.2 Using signatures as flight summaries

The aim is to summarise the vibration characteristics during a flight using a compact representation, such that it may be transmitted from the engine to the ground-based
analysis system at the end of the flight. A speed-based vibration signature provides such a compact representation; tracked order vibration amplitudes can easily be summarised in each of the $N_\omega$ speed bins used to quantise the speed axis. For example, the summary statistics in each speed bin could include the mean, the $K$ smallest, and the $K$ largest vibration amplitudes observed during a flight in that particular sub-range of shaft speed. $K$ is typically kept small, and in current practice by jet engine manufacturers, $K$ is typically equal to 3. At the end of a flight, the summary vibration amplitudes in each speed bin can be transmitted, for each of a selected number of tracked orders.

In the approach described in this chapter, we obtain the mean and the $K$ largest vibration amplitudes, $\mu = \{\mu_1\}$ and $X' = \{\max_1, \ldots, \max_K\}$, respectively, after the first flight, for each speed bin. After $r$ flights, we will have $\mu = \{\mu_1, \ldots, \mu_r\}$ and $X' = \{\max_1, \ldots, \max_{rK}\}$. Using these summary statistics in each speed bin, we require a method of setting a novelty threshold $x_\kappa$, such that vibration amplitudes $|x|$ acquired during subsequent flights are classified “abnormal” if $|x| > x_\kappa$. If the distributions of vibration amplitudes in each speed bin are treated independently of each other, this is a set of $N_\omega$ independent univariate problems for each vibration signature.

Figure 8.2(a) shows the distribution of all vibration amplitudes occurring with a speed bin $b$ over $r = 20$ flights. It may be seen that the distribution is bimodal for this example speed bin\(^1\). Over $r = 20$ flights, with an average flight duration of 10 hours, and with a new tracked order value obtained at 5 Hz, the total number of data observed is approximately $10 \times (5 \times 3,600)$ samples per hour $= 1.8 \times 10^5$ samples.

Figure 8.2(b) shows a GMM (in black) constructed from this large quantity of vibration data using the EM algorithm described in Chapter 3. This is a multi-modal distribution, and so the EVD describing its upper tail must be found using the multi-modal $\Psi$-transformation method described in Chapter 5. The average number of data observed in each of $N_\omega = 20$ speed bins during a flight is $1.8 \times 10^5$ samples / 20 speed bins $= 9 \times 10^3$ samples per speed bin. The figure shows the EVD (in blue) corresponding to this, which shows the distribution of the maximum of these $m = 9 \times 10^3$ samples. A novelty threshold has been set at $P_e(\Psi) = 0.99$, and is shown by the red dashed

\(^1\)Engine experts believe that the bimodal distribution of vibration amplitudes in a speed bin arises because the engine is under high stress, and exhibits high operating temperatures, during periods of acceleration, while under less stress, associated with correspondingly lower temperatures, during periods of deceleration.
Figure 8.2: Setting novelty thresholds using maxima $X'$. (a) All vibration data from speed bin $b$ over $r = 20$ flights. (b) GMM constructed from the vibration data (black) and EVD determined using $\Psi$-transformation (blue). A novelty threshold set at $P_e(\Psi) = 0.99$ is shown (red dashed line). (c) The $\Psi$-transformation of $N = 10^6$ samples taken from the GMM, with the ML Gumbel distribution (red). The novelty threshold at $P_e(\Psi) = 0.99$ is shown (red dashed line). (d) Distribution of maxima $X'_b$ over $r = 20$ flights. (e) Gaussian distribution (black) fitted to $X'_b$ and its EVD (blue). A novelty threshold has been set at $P_e(x) = 0.99$. 

164
Figure 8.2(c) shows the Ψ-transformation of $N = 10^6$ extrema sampled from the GMM, computed using (5.26), and the ML Gumbel distribution $p_e(\Psi)$ fitted using Algorithm 5.1. The novelty threshold set in Ψ-space at $P_e(\Psi) = 0.99$ is shown by the red dashed line, which corresponds to the novelty threshold in data-space shown in Figure 8.2(b).

The derivation of this novelty threshold has required:

- the storage of a large number of vibration data over $r = 20$ flights;
- the construction of a GMM, using the EM algorithm applied to this large number of data;
- use of the Ψ-transformation to describe the EVD, which requires $N = 10^6$ samples (as recommended in Chapter 5); and
- determination of the ML Gumbel distribution in Ψ-space, using Algorithm 5.1, which requires iterating over the $N = 10^6$ samples.

All of these steps must be performed for each of the $b = 1 \ldots N_\omega$ speed bins. In a low-bandwidth application such as on-wing EHM, it is not possible to adopt such an approach. Instead, we derive the novelty threshold using the vibration signatures constructed from the $K = 3$ largest vibration amplitudes for each flight only.

Figure 8.2(d) shows the distribution $X'_b$ of the $K = 3$ largest vibration amplitudes in speed bin $b$ over $r = 20$ flights. There are $r \times K = 60$ data in $X'_b$, compared to the $N \approx 1.8 \times 10^5$ data in the original distribution shown in Figure 8.2(a). We will examine the shape of this distribution for various datasets in Chapter 9 and show that it is approximately a unimodal Gaussian distribution. The EVD describing the tail of a univariate, unimodal Gaussian can be found using classical EVT, as described in Chapter 4. Figure 8.2(e) shows the Gaussian distribution of $X'_b$ (in black) and the EVD (in blue) obtained using classical EVT with equations (4.5) and (4.6), based on the $m = K = 3$ maxima obtained from each flight. The novelty threshold is set at $P_e(\Psi) = 0.99$, as shown by the red line.

The derivation of this novelty threshold using $X'_b$ has avoided the need to store large quantities of vibration data. We have used univariate, unimodal EVT to compute
the EVD, for which classical EVT is suitable, and which requires no computationally-intensive operations such as GMM training, Ψ-sampling, or ML Gumbel estimation.

We note that this approach of determining a threshold using the tail of a distribution of summary extrema is common in the classical EVT literature. Coles (2001) shows how a threshold on annual rainfall may be determined using the $K$ largest rainfall values each month (where $K$ is small). Castillo et al. (2005) show similar examples using annual wind-speed data and annual flood data. Smith (1989) describes the use of the $K = 3$ maxima from a number of time periods to form a distribution from which an EVD is then determined, using data from monthly measurements of ozone levels.

The remainder of this chapter considers the use of EVT for determining the novelty thresholds $x_κ$ in each speed bin using the distribution $X_κ'$.

8.2.3 Using EVT for on-wing engine health monitoring

In on-wing EHM, we will receive $K = 3$ vibration amplitude maxima in each speed bin, after each flight, as described earlier. In the work described in previous chapters, we have taken a maximum likelihood (ML) approach to estimating EVDs. However, a disadvantage with methods that use ML estimates of the underlying data distribution, $D$, is that these estimates may be poor for small numbers of observed data, as shown in Figure 8.3 where the mean and standard deviation of a set of samples $X = \{x_1, \ldots, x_N\} \sim N(0, 1)$ are plotted for increasing $N$. It can be seen that there are errors of approximately $\pm 0.1$ in the estimate of $\mu$ and $\sigma$ until $N = 12$ samples have been observed, which corresponds to four flights in our EHM system that uses $K = 3$ maxima per flight. As shown in the figure, errors only become negligible after approximately $10^3$ data are observed, which corresponds to over 300 flights in our EHM system where only the $K = 3$ maxima in each speed bin, after each flight, are considered.

We have maximal uncertainty in our estimate of $D$ for low numbers of observed data (as shown in Figure 8.3) during the initial flights. This could lead to false-positive misclassifications, as “normal” data could easily fall outside this initial, potentially oversensitive threshold. Thus, we need a method that takes into account this uncertainty in our model when low quantities of data are available, as typically occurs in the initial use of the low-bandwidth EHM system considered in this chapter.
Figure 8.3: Mean and standard deviation of an example dataset $X = \{x_1, \ldots, x_N\}$, distributed $\sim N(0, 1)$, for increasing $N$ (log scale on $x$-axis). While $\mu \to 0$ and $\sigma \to 1$ as $N \to \infty$, the values of $\mu$ and $\sigma$ oscillate for low $N$, because the data are being generated randomly from the underlying distribution.

### 8.3 Bayesian EVT

#### 8.3.1 Introduction

In Chapter 4, we showed that it was undesirable for novelty thresholds $x_\kappa$ to increase without bound as more training data are collected. In the previous section, we also showed that our uncertainty in our estimate $\hat{D}$ of the underlying function $D$ assumed to have generated the “normal” data was at a maximum when we have only observed a small number of data. This latter problem is of particular concern for low-bandwidth applications for which the quantity of training data available for training will always be limited.

We propose here a Bayesian formulation of EVT for determining EVDs for a univariate Gaussian distribution that overcomes these disadvantages. It allows novelty thresholds to be set that do not increase without bound when observing new data, and such that uncertainty in the estimate of $D$ is incorporated into the model. We will show that this method results in novelty thresholds that are maximally conservative for small numbers of observed data, when our uncertainty in our estimate of $D$ is at its maximum, and which then become more sensitive as more data are observed, as our uncertainty in our estimate of $D$ decreases.
8.3.2 Previous Bayesian approaches to EVT

There are few previously-published applications of Bayesian methods to EVT. Several come from the more established field of component reliability, also known as lifing (Basu 1964, Holla 1966, Bhattacharya 1967), and use the three-parameter Weibull distribution (4.3). Such methods typically involve intractable integration, and proposed methods rely on quadrature approximation (Smith & Naylor 1987), Bayes linear estimates (Sinha & Sloan 1988), Laplace approximation (Singpurwalla & Song 1988), or Gibbs sampling (Dey & Lee 1992, Berger & Sun 1993) to provide numerical approximations.

Coles & Powell (1996) assessed the viability of using expert knowledge to inform the selection of prior distributions for the GEV case (4.7). In order to overcome the intractable Bayesian computation of fusing expert prior knowledge of annual rainfall measurements into a model, Gibbs sampling was employed.

Bayesian approaches have also been employed in the peaks-over-threshold method, in which the generalised Pareto distribution (GPD)

\[ H(x|c,d,\xi) = \begin{cases} 1 - (1 + \xi y)^{-1/\xi} & \text{if } \xi \neq 0 \\ 1 - \exp(-y) & \text{if } \xi = 0 \end{cases} \]  

is used to characterise the distribution of data lying above some upper threshold (Bermudez et al. 2001, Behrens et al. 2004). As described in Chapter 4, this method requires selection of this upper threshold to which the resultant EVD is sensitive.

Approaches considering the Gumbel distribution have provided Bayesian formulations for the location parameter, \( c \), while treating the scale, \( d \), as a “nuisance” parameter (Rodrigues & Louzada-Neto 1990).

8.3.3 Overview of method

As with previous applications of EVT for novelty detection in this thesis, we consider data drawn from a Gaussian distribution \( X \sim N(\mu, \lambda^{-1}) \), the tails of which are described by the Gumbel EVD. Note that we will use precision \( \lambda = \sigma^{-2} \) for later convenience. As described in the previous section, the uncertainty in our model comes from uncertainty in our estimate of \( D \). This section describes how we can build this uncertainty into our model using Bayesian techniques, to provide an EVD that is robust to errors in our estimate of \( D \), and which will thus provide robust novelty thresholds.
Previous methods have assumed that the distribution $D$ is obtained by ML estimation; i.e., that the observed data $X$ have mean and precision $\theta_{\text{ML}} = (\mu_{\text{ML}}, \lambda_{\text{ML}})$. Here, we consider all possible pairs $\theta = (\mu, \lambda)$, and define the probability of observing each pair given the data to be $P(\theta|X)$. While the ML parameters are, by definition, those for which $P(\theta|X)$ is maximised, and which are the most likely estimates of the true parameters, the probability of many other candidate model parameters $\theta$ being estimates of the true parameters may be significant, particularly for low numbers of observed data, $N$. In the proposed method, the uncertainty in $\theta$ is modelled using a Bayesian methodology; we then integrate over all values of $\theta$ to find an overall EVD.

### 8.3.4 Non-standardised data

In order to consider various pairs $\theta = (\mu, \lambda)$, we must first modify the original Gumbel equations to use non-standardised data, $X = \{x_1, \ldots, x_N\} \sim \mathcal{N}(\mu, \lambda^{-1})$. We standardise the data,

$$x' = \sqrt{\lambda}(x - \mu)$$

then compute the reduced variate using these standardised data,

$$y = \frac{x' - c}{d}$$

We then scale the EVD (4.9) by $\sqrt{\lambda}$ such that it integrates to unity, which then becomes the EVD for that distribution $D$ which has parameters $\theta = (\mu, \lambda)$:

$$p_e(x|\theta) = \frac{\sqrt{\lambda}}{d} \exp \left( -y - \exp(-y) \right)$$

(8.4)

Furthermore, we may determine the location of novelty threshold $x_\kappa$ in the upper tail of $D$, set at some probability $P(x|\theta) = \kappa$, by integrating (8.4) with respect to $x$, substituting in (8.3) and (8.2), and using the definition of $c$ from (4.5) and $d$ from (4.6) to give

$$x_\kappa = \mu + (2\lambda \ln m)^{-2} \left\{ 2 \ln m - \ln(-\ln \kappa) - \frac{\ln \ln m + \ln(4\pi)}{2} \right\}$$

(8.5)

### 8.3.5 Modelling parameter uncertainty

Our goal is to determine an overall EVD given the data, $p_e(x|X)$. Rather than considering just the ML parameters $\theta_{\text{ML}} = (\mu_{\text{ML}}, \lambda_{\text{ML}})$, we will consider $S$ pairs of possible
parameters $\theta = (\mu, \lambda)$. Each of the $S$ parameter pairs defines a potential distribution $\hat{D}$ from which the observed data $X$ may have been generated. As we observe more data, we expect that the range of potential distributions likely to have generated $X$ will decrease. By definition, after observing an infinite number of data, our estimate $\hat{D}$ will converge to the ML distribution, $(\mu_{ML}, \lambda_{ML})$.

Each of the $\hat{D}$ distributions defined by the $i = 1 \ldots S$ parameter pairs, $\theta_i = (\mu_i, \lambda_i)$, has an associated EVD describing its tails, $p_e(x|\theta_i)$. We define the overall EVD, $p_e(x|X)$, to be a weighted sum of these $S$ individual EVDs:

$$p_e(x|X) = \sum_{i=1}^{S} w_i p_e(x|\theta_i)$$  \hspace{1cm} (8.6)

where weight $w_i$ is the probability of the $i^{th}$ parameter pair $\theta_i$ being the true parameters of the actual data distribution, $D$, which is given by

$$w_i = P(\theta_i|X)$$  \hspace{1cm} (8.7)

Combining (8.6) and (8.7), we have the overall EVD,

$$p_e(x|X) = \sum_{i=1}^{S} P(\theta_i|X) p_e(x|\theta_i)$$  \hspace{1cm} (8.8)

If we consider all possible $\theta = (\mu, \lambda)$ pairs, we generalise the above to give the integral

$$p_e(x|X) = \int_\theta p_e(x|\theta) p(\theta|X) d\theta$$

$$= \int_{-\infty}^{\infty} \int_{0}^{\infty} p_e(x|\mu, \lambda) p(\mu, \lambda|X) d\lambda d\mu$$  \hspace{1cm} (8.9)

where $P(\theta|X)$ has become a joint probability distribution over $\theta = (\mu, \lambda)$.

The first integrand $p_e(x|\mu, \lambda)$ in (8.9) is the EVD given in (8.4); we now require an expression for the second integrand $p(\mu, \lambda|X)$, our uncertainty in the model parameters, $\theta$.

### 8.3.6 Modelling uncertainty in model parameters

Using Bayes’ theorem (Bishop 1995), we can express the joint distribution over the parameters as a posterior distribution,

$$p(\theta|X) \propto p(X|\theta) p(\theta)$$  \hspace{1cm} (8.10)
in terms of the data likelihood, \( p(X|\theta) \), and a prior distribution, \( p(\theta) \), that encapsulates our “best initial estimate” of \( \theta \) before we have observed data \( X \).

We may be able to make such an initial estimate of \( \theta \) based on prior expert knowledge of the system. Thus, \( p(\theta) \) can be viewed as a \textit{generic} model of the system, while \( p(\theta|X) \) can be said to be a \textit{specific} model after observing actual system data.

If we have no prior knowledge of \( \theta \), we can set \( p(\theta) \) to be the bivariate uniform distribution representing maximum uncertainty, where \( \mu \in \mathbb{R}, \lambda \in \mathbb{R}^+ \). As data are observed, uncertainty in our estimates of \( \theta \) decreases, and the posterior distribution becomes more sharply peaked around the true parameter values. Ultimately, after observing infinite data, the posterior distribution will be a Dirac delta function at \((\mu_{ML}, \lambda_{ML})\), representing absolute certainty in the model parameters.

\textbf{Recursive Bayesian learning}

Noting that \( X = \{x_1, \ldots, x_N\} \), i.e. a time-series of \( N \) data points, we may express (8.10) by separating out the contribution of \( x_N \) to give an update equation after observation of the \( N^{th} \) sample,
\[
p(\theta_N|x_1, \ldots, x_N) \propto p(x_N|\theta_{N-1}) p(\theta_{N-1}|x_1, \ldots, x_{N-1})
\]  
(8.11)
i.e., the posterior distribution after observing \( N \) data is proportional to the product of the likelihood of sample \( x_N \) and the previous posterior distribution (obtained after observing \( N - 1 \) data). Note that the first posterior distribution is given by
\[
p(\theta_1|x_1) = p(x_1) p(\theta_0)
\]  
(8.12)
where \( p(\theta_0) \) is the prior distribution \( p(\theta) \). This is \textit{recursive Bayesian learning}, and allows us to store only the current posterior distribution rather than retaining the whole data set, \( X \).

\textbf{Conjugate priors}

In order for the posterior distribution from iteration \( N - 1 \) to become the prior distribution for iteration \( N \), we must choose a functional form that is \textit{reproducing}; i.e., after multiplication with the data likelihood, the functional form of the posterior distribution must be the same as the functional form of the prior distribution. This reproducing form is a \textit{conjugate prior} (Bishop 2006).
When data are assumed to be distributed $X \sim N(\mu, \lambda^{-1})$, it can be shown (Webb 2002) that if $\lambda$ is known, the conjugate prior $p(\mu)$ takes the form $N(\mu|\mu_0, \lambda_0^{-1})$. Here, $\mu_0$ and $\lambda_0$ are hyperparameters that determine the location and scale of the Gaussian conjugate prior.

Similarly, if $\mu$ is known, the conjugate prior $p(\lambda)$ takes the form of the gamma distribution (Webb 2002),

$$\text{Gam}(\lambda|a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \lambda^{a_0-1} \exp(-b_0\lambda)$$  \hspace{1cm} (8.13)

where the gamma function is

$$\Gamma(a_0) = \int_0^\infty t^{a_0-1} e^{-t} dt$$  \hspace{1cm} (8.14)

and where $a_0$ and $b_0$ are hyperparameters determining the shape and scale of the gamma distribution.

For the general case of unknown $\mu$ and $\lambda$, we must determine the form of the conjugate prior and the associated hyperparameters. Bishop (2006) provides a derivation of the functional form of the prior distribution, which we extend to derive expressions for its hyperparameters in Appendix B. Here, we present the final functional form of the conjugate prior. This Normal $\times$ gamma form is a combination of the previous two cases, in which the conjugate prior took the Normal form when $\mu$ is unknown, and the gamma form when $\lambda$ is unknown:

$$p(\mu, \lambda) = N(\mu|\mu_0, (\beta_0\lambda)^{-1}) \text{ Gam} \left( \lambda|a_0, b_0 \right)$$  \hspace{1cm} (8.15)

This is a Normal-gamma distribution, with hyperparameters $\mu_0, \beta_0, a_0, b_0$, an example of which is shown in Figure 8.4. We may write $p(\mu, \lambda) = p(\mu|\lambda) p(\lambda)$, from which, by comparison with (8.15), it may be seen that the conjugate prior joint distribution has:

- mean $\mu$, which has a Gaussian distribution with mean $\mu_0$ and precision which is a function of $\lambda$,
- precision $\lambda$, which has a gamma distribution, with shape $a_0$ and scale $b_0$.

**Example of learning with uncertainty**

Figure 8.5 shows an example of learning the joint posterior distribution $p(\mu, \lambda|X)$ for increasing numbers of observed data, $X$. An example prior distribution (i.e., where
Figure 8.4: An example conjugate prior distribution with functional form 
\[ p(\mu, \lambda) = p(\mu|\lambda) p(\lambda) = N(\mu|\mu_0, (\beta_0\lambda)^{-1}) \text{ Gam}(\lambda|a_0, b_0) . \]

\( N = 0 \) is shown in the upper-left plot, using example hyperparameters \( \{\mu_0 = 2, \beta_0 = 1, a_0 = 5, b_0 = 6\} \). As more data are observed, the posterior joint distribution becomes increasingly peaked at the actual parameters of the underlying distribution, \( \mu = 0, \lambda = 1 \), ultimately forming the Dirac delta function at that location for \( N = \infty \).

For each posterior distribution, five example parameter pairs \( \theta = (\mu, \lambda) \) have been selected, shown as coloured circles. Each selected pair corresponds to a data distribution, \( \hat{D} \), that may have generated the observed data \( X \) with probability \( P(\theta|X) > 0.05 \), which thus represent distributions that are likely to be the true underlying data distribution, \( D \). These selected distributions are shown in Figure 8.6.

Note that the ML parameter pair is the peak of the joint posterior distribution (circled in dark blue). The corresponding ML distribution is plotted in dark blue in Figure 8.6. It can be seen that, for small numbers of observed data, these selected distributions can be very different to the ML distribution. Considering only the ML distribution is likely to be a poor estimate of the actual underlying data distribution for small values of \( N \). As more data are observed, it can be seen that the selected distributions tend towards the ML distribution.
Figure 8.5: Example of Bayesian recursive learning, showing $p(\mu, \lambda | \mathbf{X})$ for increasing numbers of observed data, $\mathbf{X}$. From upper-left to lower-right, $N = 0, 1, 5, 10, 50, 100$, respectively, where samples are shown plotted on the $x$-axis (red crosses). Five selected parameters pairs $\theta = (\mu, \lambda)$ are shown by coloured circles in each plot, corresponding to five likely distributions. N.B., $N = 0$ is the prior distribution, using example hyperparameters $\{\mu_0 = 2, \beta_0 = 1, a_0 = 5, b_0 = 6\}$.

Figure 8.6: Selected distributions corresponding to the five selected parameter pairs $\theta = (\mu, \lambda)$ in each plot of Figure 8.5. From upper-left to lower-right, $N = 0, 1, 5, 10, 50, 100$, respectively, where samples are shown plotted on the $x$-axis (red crosses).
8.3.7 Using uncertainty to determine an EVD

We now have an expression for the posterior distribution \( p(\mu, \lambda | X) \), the second term in the integral defining the overall EVD (8.9),

\[
p_e(x | X) = \int_{-\infty}^{\infty} \int_{0}^{\infty} p_e(x | \mu, \lambda) p(\mu, \lambda | X) \, d\lambda \, d\mu
\]

This integral is of the form \( f(x, y) = \exp\{Ax + By + Cxy + D + \exp(Ex + Fy + G)\} \), for which no closed solution can be found. However, using (8.8), we may take advantage of the known form of \( p(\mu, \lambda | X) \) to perform a numerical approximation by considering a limited number \( S \) of parameter pairs \( \theta = (\mu, \lambda) \) from the posterior distribution.

While other numerical approximations could be used, such as a Laplacian approximation (Bishop 2006), we note that knowing the closed form of the distribution allows us to form an accurate, specific sampling solution. In particular, the marginal distribution of the precision \( \lambda \) is not symmetric, and so a Laplacian approximation, in which a Gaussian distribution is used to approximate the joint density, would not accurately model the tails of \( p(\mu, \lambda | X) \).

**Sampling the posterior**

We define a \( S \times S \) mesh of samples over the joint posterior distribution \( p(\mu, \lambda | X) \), such that each sample represents an equiprobable area in the \( (\mu, \lambda) \) plane; i.e., we partition \( p(\mu, \lambda | X) \) into \( S \times S \) regions, each with area-under-the-curve \( 1/S^2 \). An example is shown in Figure 8.7, for \( S = 13 \).

To perform this partitioning, we will consider the \( \mu \)-axis and \( \lambda \)-axis separately, which correspond to the \( x \)- and \( y \)-axes in Figure 8.7, respectively. To find the \( S \times S \) samples:

1. Find \( S \) equiprobable samples along the plane \( \lambda = \lambda_{ML} \); i.e., the plane parallel to the \( x \)-axis passing through the peak of \( p(\mu, \lambda | X) \).

2. For each sample \( S_i \) in Step 1, find \( S \) equiprobable samples along the plane \( \mu = S_i \); i.e., the plane parallel to the \( y \)-axis passing through \( \mu = S_i \).
Figure 8.7: An example posterior distribution $p(\mu, \lambda | X)$ divided into $S \times S$ equiprobable regions (shown by red lines), with $S = 13$ here. A sample $(\mu, \lambda)$ is placed at the centroid of probability mass in each region (red dots).

**Sampling the distribution of the mean, $\mu$**

In order to sample the plane parallel to the $\mu$-axis of the posterior (Step 1, above), we must find the functional form of the plane $\lambda = K$ for some constant $K$,

$$
p(\mu, \lambda | X) \bigg|_{\lambda=K} = N \left( \mu | \mu_N, (\beta_N \lambda)^{-1} \right) \text{Gam} \left( \lambda | a_N, b_N \right) \bigg|_{\lambda=K}$$

$$
= K' \ N \left( \mu | \mu_N, (\beta_N \lambda)^{-1} \right) \tag{8.17}
$$

where

$$
K' = \frac{b_N^{a_N}}{\Gamma(a_N)} K^{a_N-1} \exp \left\{ -b_N K \right\} \tag{8.18}
$$

from which it can be seen that the equation for the plane $\lambda = K$ is the Gaussian distribution, scaled according to a function of $K$.

Using the cdf for the Gaussian distribution,

$$
P(\mu | X) = \frac{1}{2} \left[ 1 + \text{erf} \left\{ \sqrt{\frac{\beta_N \lambda}{2}} (\mu - \mu_N) \right\} \right] \tag{8.19}
$$
we can partition the distribution corresponding to the plane $\lambda = K$ into $S$ equiprobable regions, placing a sample in the centre of each, as shown in Figure 8.8.

**Sampling the distribution of the precision, $\lambda$**

In order to sample the plane parallel to the $\lambda$-axis of the posterior (Step 2, above), we must find the functional form of the plane $\mu = K$ for some constant $K$,

$$p(\mu, \lambda | \mathbf{X}) \big|_{\mu=K} = N(\mu | \mu_N, (\beta_N \lambda)^{-1}) \text{Gam}(\lambda | a_N, b_N) \big|_{\mu=K}$$

This will not take the form of a simple gamma distribution, because the Gaussian distribution is dependent on $\lambda$. Expanding (8.20),

$$p(\mu, \lambda | \mathbf{X}) \big|_{\mu=K} = \frac{\beta_N^{1/2}}{(2\pi)^{1/2}\Gamma(a_N)} \lambda^{1/2} b_N^{a_N} \lambda^{a_N-1} \exp \left\{ - \left( b_N + \frac{\beta_N}{2} (K - \mu_N)^2 \right) \lambda \right\}$$

$$= \frac{\beta_N^{1/2}}{(2\pi)^{1/2}\Gamma(a_N)} \lambda^{1/2} [Q c_N^{a_N}] \lambda^{a_N-1} \exp \{-c_N \lambda\}$$

$$= K' \sqrt{\lambda} \text{Gam}(\lambda | a_N, c_N)$$

where $c_N = b_N + \frac{\beta_N}{2} (K - \mu_N)^2$ and $K' = Q \left( \frac{\beta_N}{2\pi} \right)^{1/2}$, for some normalising constant, $Q$. Formulae for $Q$ are derived in Appendix C.

Thus, the planes for fixed $\mu$ are distributions of the form $\sqrt{\lambda} \text{Gam}(\lambda)$, as shown in Figure 8.9. Such a distribution does not lend itself to being approximated by a Gaussian distribution (particularly in its tails), which motivates our use of a sample-based approximation that takes into account our knowledge of the functional form of the $\lambda$ marginal distribution.

With the equation for planes $\mu = K$ now fully specified in (8.21), (C.3), and (C.4), we may now perform equiprobable sampling for $\lambda$, as shown in Figure 8.10, using the cdf corresponding to (8.21), $P(\mu, \lambda | \mathbf{X}) \big|_{\mu=K} = 1 - \frac{\Gamma(a_N+1/2, c_N \lambda)}{\Gamma(a_N+1/2)}$.

**Evaluating the final EVD**

The sample mesh can be created rapidly using the above method, and thus we may use (8.8), where the first term $\forall i$, $P(\theta_i | \mathbf{X}) = 1/S^2$, and the second term $p_e(x | \theta_i)$ is the EVD corresponding to the distribution parameterised by the $S$ samples, $\theta_i = (\mu_i, \lambda_i)$, which is given by (8.4); i.e.,

$$p_e(x | \mathbf{X}) = \sum_{i=1}^{S^2} \frac{P(\theta_i | \mathbf{X}) p_e(x | \theta_i)}{\text{Parameter probabilities Ind. EVDs}} \tag{8.22}$$
Figure 8.8: Dividing $P(\mu|X)$ into $S = 13$ equiprobable regions (upper plot), with a sample placed in each at the centroid of probability mass. The corresponding Gaussian $p(\mu|X)$ is shown in the lower plot, showing the $S$ samples placed using $P(\mu|X)$. 
For it to be a close approximation to the full integral in (8.9), the total probability mass enclosed by the $S^2$ regions in the sample mesh must be close to unity. Table 8.1 shows the total integrated probability mass obtained by integrating the $S^2$ regions of an example posterior distribution. To perform this integration, we have used an adaptive, recursive implementation using Simpson’s rule (Mathworks 2008). It can be seen that even for small $S$, the total integrated area is close to unity.

In order to ensure that the $S^2$ weights in (8.22) sum to unity, we introduce a normalising factor,

$$p_e(x|X) = \frac{1}{\sum_{j=1}^{S^2} P(\theta_j|X)} \sum_{i=1}^{S^2} \left[ \frac{P(\theta_i|X)}{\sum_{j=1}^{S^2} P(\theta_j|X)} p_e(x|\theta_i) \right]$$

(8.23)

$$= \frac{1}{\sum_{j=1}^{S^2} P(\theta_j|X)} \sum_{i=1}^{S^2} P(\theta_i|X) p_e(x|\theta_i)$$

Figure 8.9: Distribution of a plane $\mu = K$ through the posterior $p(\mu, \lambda|X_T)$.

Table 8.1: Total probability mass obtained by integrating over $S^2$ regions in an example posterior distribution $p(\mu, \lambda|X)$, for increasing $S$.

| $S$ | $\sum P(\theta_i|X)$ |
|-----|----------------------|
| 5   | 0.9950               |
| 7   | 0.9966               |
| 13  | 0.9983               |
| 25  | 0.9991               |
| 51  | 0.9996               |
| 101 | 0.9998               |
| 201 | 0.9999               |
Figure 8.10: Dividing $P(\lambda|X)$ into $S = 13$ equiprobable regions (upper plot), with a sample placed in each at the centroid of probability mass. The corresponding pdf, $p(\mu|X)$, is shown in the lower plot, showing the $S$ samples placed using $P(\lambda|X)$. 
8.3.8 Evaluation of Bayesian EVT using synthetic data

Before applying the method to engine data in Chapter 9, we here compare the performance of the proposed Bayesian EVT method, which we term \textit{B-EVT}, with the ML method described in Chapter 4, which we term \textit{ML-EVT}. Using an artificial example dataset $X \sim N(0,1)$, EVDs were computed using both methods for increasing numbers of samples $N$.

For the purposes of illustration using synthetic data, here we fix $m$ at some constant value, as we have no test set in this example and wish only to examine the effect of training on the resultant EVDs. Selection of $m$ will be discussed in the next chapter, and here we here choose $m = 3$. A sampling mesh $S \times S = 7 \times 7$ was used for B-EVT.

Figure 8.11 shows EVDs describing the upper tail of the underlying distribution $D$, obtained using both methods for selected values of $N$. EVDs obtained using the ML-EVT method show no consistent trend in either location\(^2\) or scale\(^3\): the distribution for $N = 1$ is centred on a lower value of $x$ than the distribution for $N = 3$, yet the distribution for $N = 15$ is located even lower. This is due to the error in the ML estimate of the distribution parameters for small numbers of data, as was shown in Figure 8.3.

For $N < 100$, it can be seen that the peak of the EVD varies over approximately $1\sigma$, noting that the $x$-axis has units $1\sigma$ because $X \sim N(0,1)$. This shows that EVDs computed using the ML-EVT method are particularly vulnerable to poor estimation of parameters for small $N$.

Note from the lower plot in Figure 8.11 that, while the EVDs obtained using ML-EVT vary in scale in a non-monotonic manner (as can be seen from the fact that their heights are neither constantly increasing nor decreasing with increasing $m$), location of the novelty thresholds set in the tail of the EVDs do tend to increase monotonically, as shown previously in Figure 4.5. Oscillations in the peaks of the EVDs are caused by the variations associated with sampling small numbers of data from $D$.

In comparison, EVDs obtained using the B-EVT method start with high values of

\(^2\)The location of a distribution is conventionally defined to be its translation in the direction of the $x$-axis. For the Gaussian distribution, the location parameter is $\mu$, and for the Gumbel distribution, the location parameter is $c$.

\(^3\)The scale of a distribution is its scaling in the direction of the $x$-axis. For the Gaussian distribution, the scale parameter is $\sigma$, and for the Gumbel distribution, the scale parameter is $d$. 181
Figure 8.11: Overall EVD computed from $N$ artificial data, for increasing $N = 1, 3, 5, 6, 15, 100$, using B-EVT (upper plot) and ML-EVT (lower plot). $x$ and $p_e(x|\mathbf{X})$ are shown on the $x$- and $y$-axes, respectively.

Figure 8.12: Overall EVD computed from $N$ artificial data, for increasing $N$, using B-EVT (shown in red) and ML-EVT (shown in black). $x$ and $p_e(x|\mathbf{X})$ are shown on the $x$- and $y$-axes, respectively.

location and scale, due to the high uncertainty in the estimate of the underlying data distribution for small $N$. These values decrease as more data are observed (more so in the case of the location), corresponding to decreased uncertainty in the parameter estimates.

Note that the EVDs obtained using both methods are similar for $N = 100$, indicating that the Bayesian solution converges to the ML solution for large $N$. Figure 8.12 shows EVDs obtained from B-EVT and ML-EVT plotted on the same axes, for increasing $N$. The convergence of the B-EVT and ML-EVT solutions is shown, and the decrease in
location and scale of the B-EVT solution (its peak is shifted downwards on the x-axis, and its spread is decreased) is also evident. This corresponds to decreasing uncertainty in the estimate of the underlying distribution assumedly responsible for generating the observed data.

8.3.9 Setting novelty thresholds

In Section 8.3.4, we formulated an expression (8.5) for the location of the novelty threshold \( x_\kappa \) for an individual EVD. In order to determine the location of novelty thresholds using the overall EVD obtained from B-EVT, we must integrate (8.23) to give the corresponding cdf,

\[
P_e(x|X) = \int p_e(x|X) \, dx
\]

\[
= \int \frac{1}{Q} \sum_{i=1}^{S^2} P(\theta_i|X) \, p_e(x|\theta_i) \, dx
\]

\[
= \frac{1}{Q} \sum_{i=1}^{S^2} P(\theta_i|X) \int p_e(x|\theta_i) \, dx
\]

\[
= \frac{\sqrt{\lambda}}{Q} \sum_{i=1}^{S^2} P(\theta_i|X) \exp\{ - \exp(-y) \}
\]

for normalising factor \( Q = \sum_{i=1}^{S^2} P(\theta_i|X) \).

Figure 8.13 shows the novelty threshold \( x_\kappa \) obtained using (8.24) for B-EVT and (8.5) for ML-EVT, with \( P_e(x|X) \leq 0.999 \). Standardisation of the data using (8.3) results in small oscillations for both curves, which correspond to the variability in the estimation of the underlying data distribution for small \( N \).

It can be seen that the novelty threshold for ML-EVT is initially very low, ultimately converging to \( x_\kappa = 5.1 \) when \( N = 100 \). Conversely, the novelty threshold for B-EVT takes high values for low \( N \), converging towards the ML solution at approximately \( N = 100 \). Thus, a conservative novelty threshold is set when data uncertainty is high (for low \( N \)), which then becomes less conservative as data uncertainty decreases, as desired.
Figure 8.13: Novelty threshold $x_\kappa$ set at $P_\kappa(x|X) \leq 0.999$ for B-EVT (shown in red) and ML-EVT (shown in black), for increasing $N$.

8.4 Conclusions

We have described how vibration signatures may be used to construct a model of normality, such that test data gathered with an on-wing EHM system during new flights may be compared with it in order to identify novelty. This requires the setting of novelty thresholds on tracked order vibration amplitude within each speed bin, which is performed in a principled manner using EVT. When using signatures that summarise tracked order vibration amplitudes for the entire flight, there are initially few data points available to construct the model of normality (e.g., during the first 50 flights, which corresponds to the first 150 data points in each speed bin). Hence EVT has been set within a Bayesian framework to incorporate uncertainty into the model in a principled manner.

This gives novelty thresholds which are conservative when uncertainty in the model of normality is at its highest (i.e., during the first few flights). As more data are observed, and uncertainty in our model decreases, novelty thresholds become less conservative, converging towards the ML solution.

In Chapter 9, we will use this technique for novelty detection in tracked-order signatures, and compare it with existing heuristic approaches and with ML EVT methods.
Chapter 9

Low-Bandwidth Novelty Detection with Tracked Order Data

9.1 Introduction

In Chapter 8, we described the use of speed-based vibration signatures as a compact representation summarising the vibration amplitude of a tracked order throughout a flight, which may be transmitted from the aircraft to a ground-based monitoring system for analysis. This chapter describes an investigation in which the Bayesian EVT (B-EVT) techniques described in Chapter 8 are used for novelty detection using such signatures as inputs. The performance of the proposed algorithm is compared with existing techniques, and with the current practice of engine experts.

9.2 Analysing Vibration Signatures

9.2.1 Current practice

Speed-based tracked order vibration signatures are currently analysed using heuristic methods (King et al. 2002). As described in Chapter 8, the vibration signature for a single flight comprises the mean and the $K$ largest vibration amplitudes, $\mu_b$ and $X'_b = \{\text{max}_{b,1}, \ldots, \text{max}_{b,K}\}$, respectively, occurring in each of $b = 1 \ldots N_\omega$ speed bins. In current practice, $K = 3$ and $N_\omega = 20$. The total number of data $N_b$ observed in each speed bin during the flight is also recorded.

These signatures are collected for a number of flights $r$, giving a set of data $\{\mu_b, X'_b\}$ for each speed bin $b$, where $\mu_b = \{\mu_{b,1} \ldots \mu_{b,r}\}$ and $X'_b = \{\text{max}_{b,1}, \ldots, \text{max}_{b,r \times K}\}$. As described in Chapter 8, we wish to set a novelty threshold in the tail of the distribution of $X'_b$. 
The current practice of engine experts is to set thresholds $x_{b,\kappa}$ in speed bin $b$ using:

$$x_{b,\kappa} = \bar{\mu}_b + K_h \left[ \max(X_b') - \bar{\mu}_b \right]$$

(9.1)

where $K_h$ is some constant (typically $1.0 \leq K_h \leq 2.0$), and where $\bar{\mu}_b$ is the mean vibration amplitude in speed bin $b$ over the $r$ flights:

$$\bar{\mu}_b = \frac{1}{\sum_i N_{b,i}} \sum_{i=1}^r N_{b,i} \mu_{b,i}$$

(9.2)

i.e., the mean $\bar{\mu}_b$ is calculated using the number of data $N_{b,i}$ in speed bin $b$ for flight $i$ and the corresponding mean $\mu_{b,i}$.

The novelty thresholds are therefore set at some multiple $K_h$ of the extremum-to-mean distance. The example from Figure 8.2 is shown again in Figure 9.1(a), in which a histogram of the vibration amplitudes within a speed bin over $r = 20$ flights is shown. The mean of these vibration amplitudes is shown as a green line at $x = 0.13$ in $s^{-1}$, calculated from the $r = 20$ vibration amplitudes in that bin using (9.2). The maximum vibration amplitude observed is shown as a black line at $x = 0.24$ in $s^{-1}$, and a novelty threshold $x_{\kappa} = 0.295$ has been set using (9.1) with $K_h = 1.5$.

As described in Chapter 8, the full set of vibration amplitudes within a speed bin cannot be stored in low-bandwidth applications. Figure 9.1(b) shows the distribution $X_b'$, formed from the $K = 3$ largest vibration amplitudes in speed bin $b$ for each of the $r = 20$ flights, as shown in Chapter 8.

The heuristic method of setting the novelty threshold using (9.1) disregards the distribution of data in a speed bin, and instead uses only two values (the maximum and mean vibration amplitudes observed in that speed bin). Hence the location of the novelty threshold will be highly sensitive to unexpected transients.

### 9.2.2 Using EVT

In this subsection, we describe how EVT can be used to set a novelty threshold using the available vibration amplitudes in each speed bin, $X_b'$. We will later show that $X_b'$ is approximately distributed according to the univariate Gaussian distribution, and thus the tails of $X_b'$ are in the domain of attraction of the Gumbel distribution. We will compare the ML method of finding this distribution, described in Chapter 4, with the Bayesian method described in Chapter 8.
Setting novelty thresholds with ML-EVT

In Chapter 4, we showed that the EVT parameter $m$ should be set according to the number of test data observed from the underlying generative distribution. In this low-bandwidth application, we obtain $K = 3$ new vibration amplitudes in each speed bin after each flight, to be tested against the current model of normality after each flight. Thus, we set $m = K = 3$ for the ML-EVT method.

Bayesian EVT

The Bayesian EVT formulation described in Chapter 8 differs from the ML-EVT method by taking into account uncertainty in the model of normality when setting a novelty threshold. However, the EVT parameter $m$ is set in the same way. Thus, as with ML-EVT, we set $m = K = 3$.

9.3 Data

The investigation described in this chapter was performed using the eight datasets detailed in Table 9.1. Each dataset contains vibration amplitudes for the 1LP, 1IP, and 1HP tracked orders.

Some of these datasets include examples of engine events, and other datasets contain entirely “normal” data throughout, as described below.
Table 9.1: Datasets used in the investigation described in this chapter.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Engine</th>
<th>Flights</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>3-shaft class A, engine I</td>
<td>57</td>
<td>$C_{1,57}$ exhibits component-release event</td>
</tr>
<tr>
<td>$C_2$</td>
<td>3-shaft class A, engine II</td>
<td>27</td>
<td>$C_{2,27}$ contains HP compressor event</td>
</tr>
<tr>
<td>$C_3$</td>
<td>3-shaft class A, engine III</td>
<td>23</td>
<td>$C_{3,17}$ contains blade-rubbing event</td>
</tr>
<tr>
<td>$C_4$</td>
<td>3-shaft class B, engine I</td>
<td>29</td>
<td>$C_{4,29}$ contains component-release event</td>
</tr>
<tr>
<td>$C_5$</td>
<td>3-shaft class A, engine IV</td>
<td>87</td>
<td>Normal engine data</td>
</tr>
<tr>
<td>$C_6$</td>
<td>3-shaft class A, engine V</td>
<td>96</td>
<td>Normal engine data</td>
</tr>
<tr>
<td>$C_7$</td>
<td>3-shaft class A, engine VI</td>
<td>72</td>
<td>Normal engine data</td>
</tr>
<tr>
<td>$C_8$</td>
<td>3-shaft class A, engine VII</td>
<td>126</td>
<td>Normal engine data</td>
</tr>
</tbody>
</table>

Event data

Dataset $C_1$ has evidence of a component-release failure during the last of the 57 flights. Engine experts have noted from retrospective analysis of the data in $C_1$ that the component became loose during flight $C_{1,49}$, and stayed loose in the engine for the next 7 flights, until it escaped into the gas-path during flight $C_{1,57}$, causing a major engine event, identifiable in the 1LP data. The loose component was only identified retrospectively, and only the final engine event was observed in real time.

Dataset $C_2$ contains an event related to the compressor of the HP shaft that caused a major engine event in its final flight, $C_{2,27}$. Retrospective data analysis by engine experts has shown that a precursor event occurred during flight $C_{2,23}$, identifiable in the 1HP data. Again, only the final engine event was observed at the time.

Dataset $C_3$ shows evidence of an event during flight $C_{3,17}$ in which it is believed that engine fan blades impacted with the engine casing, causing minor damage. This event was only identified by retrospective analysis by engine experts, who noted that the engine continued to run for a further 6 flights before it was taken off-wing, and that the event could be identified in the 1LP vibration data of flight $C_{3,17}$.

Dataset $C_4$ contains a second example of component release, which was detected during the final flight of the engine $C_{4,29}$. Retrospective analysis showed that the component became significantly damaged during the preceding flight, $C_{4,28}$, an occurrence which is again identifiable in the 1LP vibration data.
Figure 9.2: Distributions of maxima $X'_b$, where $b$ is the bin number in the range 1 ... 20, accumulated over large numbers of flights in “normal” datasets, taking 3 maxima per flight, normalised (by dividing all values by the area-under-the-histogram) such that they are estimates of $p(X'_b)$. (a), (b), and (c) show 1LP maxima $X'_{10}$, $X'_{15}$, and $X'_{20}$ for speed bins $b = 10$, 15, and 20, respectively, collected from all flights in dataset $C_5$; (d), (e), and (f) show 1IP maxima $X'_{10}$, $X'_{15}$, and $X'_{20}$, respectively, collected from all flights in dataset $C_6$.

“Normal” data

Datasets $C_5$ to $C_8$ were used to investigate the performance of the various algorithms with datasets consisting entirely of “normal” engine data. These are comparatively large datasets, which allow the performance of each method to be evaluated over longer periods than would otherwise be possible using the event datasets. Any detection of novelty that occurs when analysing these datasets is necessarily a false positive.

These “normal” datasets can also be used to test the assumption made in Section 9.2.2 that the distributions $X'_b$ are approximately Gaussian. Figure 9.2 shows example distributions from the “normal” datasets, which are seen to be approximately Gaussian.

More formally, we require a method of testing the hypothesis that the distribution $X'_b$ of $N$ data for some bin $b$ is Gaussian. The Kolmogorov-Smirnov (K-S) test (Stephens 1974, Press et al. 1992) is one means of testing this hypothesis. With this test, the null hypothesis is that the data $X'_b$ are drawn from the standard Gaussian distribution $N(0, 1)$. The K-S test rejects this null hypothesis if it holds with probability below some significance value $p$, typically $p = 0.05$. 
The K-S test uses the K-S distance $D_{KS}$:

$$D_{KS} = \max_{-\infty < x < \infty} \left| \hat{P}_X(x) - P_N(x) \right|$$

which is the maximum distance between the empirical cdf of the data $\hat{P}_X(x)$ and the cdf of the standard Gaussian distribution, $P_N(x) = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x}{\sqrt{2}} \right) \right]$.

In the case of the null hypothesis (i.e., that the data are drawn from the standard Gaussian distribution), the K-S distance $D_{KS}$ has a cdf according to the Kolmogorov distribution,

$$P(D_{KS}) = 1 - 2 \sum_{i=1}^{\infty} (-1)^{i-1} e^{-2i^2D_{KS}^2}$$

which gives the probability that a K-S distance $D_{KS}$ is observed under the null hypothesis. The null hypothesis is rejected, and the data thus assumed to be drawn from some distribution other than the standard Gaussian, if

$$D_{KS}\sqrt{N} > \kappa$$

where $\kappa$ is that maximum allowed K-S distance for some significance level $p$,

$$P(\kappa) = 1 - p$$

One advantage of the K-S test is that it provides accurate estimation of the “Gaussianity” of data even for small $N$ (e.g., $N = 4$). This is particularly useful for our case in which $X'_b$ contains $N = 3r$ values, for $r$ flights, which may be small.

We form distributions for all three fundamental tracked order amplitudes (1LP, 1IP, and 1HP) for each of the $b = 1 \ldots N_\omega = 20$ speed bins, for the two largest “normal” datasets, $C_6$ and $C_8$, and then standardise each $(X'_b - \mu)/\sigma$, for sample mean and standard deviation $\mu, \sigma$, respectively. These standardised distributions are then compared to the standard Gaussian $N(0,1)$ using the K-S test as described above. Resultant $p$-values for the null hypothesis are shown in Table 9.2.

The table shows that distributions from dataset $C_6$ are closer to the standard Gaussian than those from dataset $C_8$, as shown by the higher overall $p$-values for the former. Note that the null hypothesis that the data are distributed according to the Gaussian is rejected if $p < 0.05$, which occurs for some distributions of 1IP data in $C_6$, and for some distributions in 1LP and 1HP for $C_8$, indicating that there are some distributions
Table 9.2: Results of performing the K-S test on distributions $X'_b$ from “normal” datasets, where the minimum, maximum, and average $p$ values are given for $N_\omega = 20$ speed bins in each case.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TO</th>
<th>min $p$</th>
<th>max $p$</th>
<th>mean $p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_6$</td>
<td>1LP</td>
<td>0.07</td>
<td>0.98</td>
<td>0.78</td>
</tr>
<tr>
<td>$C_6$</td>
<td>1IP</td>
<td>0.04</td>
<td>0.99</td>
<td>0.81</td>
</tr>
<tr>
<td>$C_8$</td>
<td>1HP</td>
<td>0.11</td>
<td>0.96</td>
<td>0.71</td>
</tr>
<tr>
<td>$C_8$</td>
<td>1LP</td>
<td>0.04</td>
<td>0.97</td>
<td>0.83</td>
</tr>
<tr>
<td>$C_8$</td>
<td>1IP</td>
<td>0.07</td>
<td>0.94</td>
<td>0.66</td>
</tr>
<tr>
<td>$C_8$</td>
<td>1HP</td>
<td>0.03</td>
<td>0.97</td>
<td>0.58</td>
</tr>
</tbody>
</table>

Table 9.3: Results of performing the K-S test on distributions $X'_b$ from “normal” datasets, where the minimum, maximum, and average $p$ values are given for $N_\omega = 20$ speed bins in each case.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total number of distributions</th>
<th>Number of distributions with $p &lt; 0.05$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_6$</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>$C_8$</td>
<td>60</td>
<td>3</td>
</tr>
</tbody>
</table>

which are non-Gaussian. Table 9.3 shows the number of distributions determined to be non-Gaussian according to the K-S test, noting that each dataset contains distributions for $N_\omega = 20$ speed bins $\times \{1LP,1IP,1HP\} = 60$ distributions. In the worst case, that of $C_8$, the number of non-Gaussian distributions is $3/60 \times 100 = 5.0\%$. This number of non-Gaussian distributions is small, and so the assumption of Gaussianity made by the EVT methods described in Section 9.2.2 can be concluded to be appropriate.

We note finally that a second method for estimating the “Gaussianity” of a dataset is the Lilliefors test (Lilliefors 1967), in which the comparison is made against a Gaussian distribution with mean and variance unknown $N(\mu, \sigma)$, rather than against the standard Gaussian distribution $N(0,1)$. With the K-S test, we standardised the data according to the ML values of $\mu, \sigma$ (i.e., the sample mean and variance) such that the K-S test could be used. While the Lilliefors test avoids making this ML assumption, it has the disadvantage that it requires Monte-Carlo estimation of the $p$-value for the null hypothesis (Conover 1980). This makes the method computationally expensive and unsuitable for estimation of the “Gaussianity” of a distribution in real-time for low-bandwidth applications, such as in this case. For the investigation described by this chapter, the Lilliefors test was performed using the same data as used previously with
the K-S test in order to verify the result obtained in Tables 9.2 and 9.3. The number of distributions rejected by the Lilliefors test for $p < 0.05$ was unchanged from those rejected by the K-S test, being 2 distributions for dataset $C_6$ and 3 distributions for datasets $C_8$.

### 9.4 Methodology

Chapter 2 described the partitioning of available data into training, validation, and test sets. Figure 9.3 shows the partitioning of the data described in Section 9.3 into these sets. As described in Chapter 2, we take an engine-specific approach to novelty detection, and so models of normality are constructed for each of the eight datasets shown in Figure 9.3. The training/validation datasets, $C_1$ and $C_2$, are used to determine the probability $\kappa$ at which novelty thresholds should be set. With $\kappa$ fixed, results are then reported using the independent “abnormal” and “normal” test datasets shown in the figure.

Chapter 8 suggested that the B-EVT method could be used to set novelty thresholds from the start of monitoring (i.e., the first flight). In practice, an alerting system would be switched on after a small number of flights. Due to the limited number of data available for model construction in a low-bandwidth approach, we first acquire data from some minimum number of flights, $r_{\text{min}}$, before constructing a model of normality. The current practice of engine experts is to use vibration signatures from a minimum of $r_{\text{min}} = 5$ flights to set novelty thresholds (Bannister et al. 2008), and so we use this value in the work described by this chapter, as shown in Figure 9.4.

### 9.5 Results

This section reports results obtained using all of the methods described in Section 9.2.2 to perform novelty detection for the eight datasets of vibration data described in Section 9.3.

#### 9.5.1 Classification

Models of normality were constructed for $N_\omega = 20$ speed bins for each of the fundamental tracked orders (1LP, 1IP, and 1HP). Novelty thresholds were set using a range of
Figure 9.3: Partitioning of the eight available datasets into training/validation and test sets. The test sets comprise two datasets containing evidence of subtle engine events (shown in red), and four datasets containing only “normal” data (shown in green).

Figure 9.4: Constructing a model of normality from flight data after a minimum number of flights $r_{\text{min}} = 5$ have been performed. Data from subsequent flights are used to update the model of normality if they are deemed to be “normal”. Here, data from flight $r = 7$ are deemed to be “abnormal”, and so the model is not updated.
candidate values for the metaparameters of each method, as described below. Data from each flight were classified with respect to the model of normality using data obtained up to the previous flight, for each of the candidate values of the metaparameters.

A classification of “normal” was made if test data in each speed bin for a given tracked order were within the novelty thresholds for that bin, for that flight. A classification of “abnormal” was made if a novelty threshold was exceeded in any speed bin for a given tracked order. We note that a classification of “abnormal” after a single bin is deemed abnormal is necessary because engine events may occur in a single speed bin, for a single tracked order, for a single flight.

These classifications are deemed true-positive (TP), false-positive (FP), true-negative (TN), or false-negative (FN) depending on their agreement with the labels supplied retrospectively for each test by engine experts, corresponding to the colour scheme adopted in Figure 9.3.

The engine events contained in the datasets used in this investigation only manifest themselves as abnormal vibration amplitudes for a single tracked order in each case. For example, evidence for the event in dataset $C_1$ was identified by the experts in the 1LP vibration data, but not in the 1IP or 1HP vibration data. For this reason, classifications were made separately for each tracked order; i.e., after a flight, each method generates a classification separately for each of the 1LP, 1IP, and 1HP tracked orders.

### 9.5.2 Data quality

In order to ensure that novelty detection is only performed using valid flight data, rather than ground-based manoeuvres, aborted take-offs, or flights with missing data, classifications were made only for those flights deemed to be “valid” according to the following criteria:

(i) data must be present in 50% of the $N_\omega = 20$ speed bins covering the range of shaft speeds $[\omega_{\text{idle}} 100\%]$;

(ii) at least $N_b > 30$ data must be observed during the flight in each of the speed bins considered in (i), from which the $K = 3$ extrema are then taken;

(iii) the flight must be longer than 30 minutes in duration.
Table 9.4: Labels for training/validation datasets. The number of “normal” and “abnormal” flights, as labelled by engine experts, is shown for each tracked order as “Nm.” and “Ab.”, respectively. Total numbers of “normal” and “abnormal” flights are given in the rightmost column.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total flights</th>
<th>Valid flights</th>
<th>1LP Nm.</th>
<th>1LP Ab.</th>
<th>1HP Nm.</th>
<th>1HP Ab.</th>
<th>Total Nm.</th>
<th>Total Ab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>57</td>
<td>51</td>
<td>42</td>
<td>9</td>
<td>51</td>
<td>0</td>
<td>144</td>
<td>9</td>
</tr>
<tr>
<td>C_2</td>
<td>27</td>
<td>23</td>
<td>23</td>
<td>0</td>
<td>23</td>
<td>2</td>
<td>67</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 9.5: Classification results for novelty detection methods with varying metaparameter sets. For B-EVT and ML-EVT, metaparameters $\theta_1, \ldots, \theta_4 = \{0.90, 0.95, 0.99, 0.999\}$, respectively, which are the values of $P_e(x)$ for setting the novelty threshold based on the EVD. For the heuristic method in (9.1), metaparameters $\theta_1, \ldots, \theta_4 = \{1.2, 1.5, 1.7, 1.9\}$, respectively, which is the value of $K_h$ for setting the novelty threshold.

<table>
<thead>
<tr>
<th>Param Set</th>
<th>B-EVT FN</th>
<th>FP</th>
<th>ML-EVT FN</th>
<th>FP</th>
<th>Heuristic FN</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_1$</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>112</td>
<td>0</td>
<td>155</td>
</tr>
<tr>
<td>$\theta_2$</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>82</td>
<td>6</td>
<td>68</td>
</tr>
<tr>
<td>$\theta_3$</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>64</td>
<td>9</td>
<td>37</td>
</tr>
<tr>
<td>$\theta_4$</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>24</td>
<td>9</td>
<td>18</td>
</tr>
</tbody>
</table>

The above values have been previously used in determining the validity of flight signatures (Clifton 2006).

### 9.5.3 Training and validation

The total number of “normal” and “abnormal” labels for each of the two training/validation datasets is given in Table 9.4.

Overall classification performance for datasets $C_1$ and $C_2$ is shown in Table 9.5, for a range of metaparameters for each method.

The table shows that the ML-EVT and B-EVT methods have equal numbers of FN classifications as the novelty threshold is varied over $P_e(x)$, with all flights with engine events being correctly classified with $P_e(x) \leq 0.99$. However, ML-EVT suffers from large numbers of FP classifications, whereas B-EVT has a much lower number, typically one order of magnitude lower than that of the ML-EVT method. The heuristic method of (9.1) has the largest number of FN classifications, and also suffers from large numbers of FP classifications.

For the case of jet engine monitoring, a FN classification is more costly than a FP
Figure 9.5: Comparison of novelty threshold \( x_\kappa \) set for event data from speed bin \( b = 4 \) of \( N_\omega = 20 \) of dataset \( C_1 \) for varying probability \( P_e(x) \). For each flight, the 3 maxima from the 1LP vibration signature are shown. Novelty thresholds for the first \( r_{\text{min}} = 5 \) flights are shown as a thick line, during which novelty detection is not performed. (a) Novelty thresholds set at \( P_e(x) = 0.90, 0.99, \) and 0.999 are shown in green, black, and red, respectively. The value of the sampling grid used by B-EVT was \( S = 7 \), as defined in Section 8.3.7. (b) Novelty thresholds set with \( S = 3, 7, 11 \) are shown in green, black, and red, respectively (\( P_e(x) = 0.99 \)).

classification, as the former indicates that a potentially important engine event may be ignored, which is typically more costly than the latter. Thus, we must select values for the metaparameters such that FN classifications are minimised, and then accept as few FP classifications as may be achieved. From the results shown in Table 9.5, we select \( P_e(x) = 0.99 \) for B-EVT and ML-EVT in order to minimise the number of FN classifications, while then accepting the lowest number of FP classifications. For the heuristic method, we must select \( K_h = 1.2 \) in order to minimise the number of FN classifications, though this results in a very high number of FP classifications.

These results will be considered more closely in the following sections, in which we will examine the ability of each method to identify each engine event, and also consider cases in which only “normal” data are present and in which FP classifications may occur.

**Dataset \( C_1 \)**

Figure 9.5(a) shows speed bin \( b = 4 \) for the 1LP tracked order, in which the maxima of each flight are plotted with novelty thresholds set using B-EVT for three values of \( P_e(x) \). Novelty thresholds for the initial \( r_{\text{min}} = 5 \) flights are shown by thick lines. Note that novelty detection is not performed during these first \( r_{\text{min}} \) flights, as described in
Section 9.4.

The figure shows that the $P_e(x) = 0.99$ novelty threshold allows correct classification of the event data occurring in the final 9 flights of the engine. The novelty threshold set at $P_e(x) = 0.90$ is less conservative, and maxima from several “normal” flights lie above it, resulting in FP classifications for those flights. The novelty threshold set at $P_e(x) = 0.999$ is the most conservative of those shown, such that the maxima from some of the final 9 flights lie beneath it, resulting in FN classifications.

As these maxima from the event data fall beneath the novelty threshold (shown by the red line for $P_e(x) = 0.999$), they are deemed to be “normal” data and added to the training set for testing novelty in the subsequent flight. This causes a significant increase in the $P_e(x) = 0.999$ novelty threshold, as these extreme data become part of the model’s description of normal 1LP vibration amplitude (behaviour not displayed by the $P_e(x) = 0.90$ and 0.99 thresholds, shown by the green and black lines, respectively).

We note also that the novelty threshold set using only the prior distribution (i.e., that which is used for novelty detection during flight $C_{1,1}$) is very high, as this corresponds to the situation in which we have no information about the engine being monitored other than our prior information (considered below).

Section 8.3.7 described the use of a sampling mesh of $S \times S$ elements for the B-EVT method. Figure 9.5(b) shows the negligible effect of varying sampling parameter $S$, which is the number of samples taken from the $p(\mu, \lambda)$ distribution when determining the final EVD of extrema within a speed bin. For $S = 3, 7,$ and 11, the resultant novelty thresholds set at $P_e(x) = 0.99$ are very similar. Throughout the investigation described in this chapter, a value of 7 for $S$ is used.

The effect of varying the prior distribution used by B-EVT is shown in Figure 9.6. Though novelty thresholds eventually converge to the same point no matter which prior distribution is used, it may be seen from the figure that using the uniform prior distribution results in a very conservative novelty threshold for the first flights (shown in red). Using a uniform prior, we have maximal uncertainty in our model parameters, corresponding to a high novelty threshold.

Using a prior distribution in which the mean and variance are taken from the mean and variance of tracked order amplitudes in this speed bin ($b = 4$) over the “normal” first
Figure 9.6: Comparison of novelty threshold $x_\kappa$ set for data from speed bin $b = 4$ of $N_\omega = 20$ of dataset $C_1$ using different prior distributions. For each flight, the 3 maxima from the 1LP vibration signature are shown. Novelty thresholds set using a prior distribution generated from available “normal” training data for this engine class are shown in black. Novelty thresholds set using the uniform prior distribution are shown in red ($P_e(x) = 0.99$).

20 flights in datasets $C_1$ and $C_2$ results in a less conservative novelty threshold (shown in black). We use this prior distribution, generated from a population of representative engines, for the remainder of the work described in this chapter.

The novelty threshold set with B-EVT at $P_e(x) = 0.99$ is compared with those set with ML-EVT in Figure 9.7(a). It may be seen that novelty thresholds obtained using ML-EVT converge to those obtained using the B-EVT method in this example.

Figure 9.7(b) shows the use of the heuristic method of (9.1) to set novelty thresholds in comparison with those obtained using B-EVT. The heuristic method is dependent only on the maximum value of the training set and the mean of the corresponding bin, as defined in (9.1), and thus the resultant novelty threshold changes only as the mean of the training data changes, or when a new maximum value is observed (as may be seen after flight $C_{1,49}$ for the case of $K_h = 1.5$ shown by the green line). It is very difficult for this method to find a compromise between generating FP classifications when observing “normal” data and being sufficiently sensitive to identify data corresponding to engine events as being abnormal.

The evolution of the novelty thresholds set using B-EVT after observing consecutive flights from dataset $C_1$ is shown in Figure 9.8(a). The novelty thresholds initially take relatively high values, reaching 0.45 in s$^{-1}$ around shaft speed $\omega_{LP} \approx 90\%$, then decrease
as uncertainty in the model of normality decreases with data from more flights being observed. Data from the engine events occurring after flight $C_{1,49}$ may be seen to exceed the novelty thresholds in speed bins $b = 4, 5, 10, 11, \text{ and } 12$, which are circled in red in the figure.

Figure 9.8(b) shows that the maximum observed vibration amplitude for this engine was $|1LP| \approx 3.0$ in s$^{-1}$, in speed bin $b = 4$. Conventional engine monitoring systems are typically set to detect vibration amplitudes above $|1LP| = 0.5$ in s$^{-1}$ (King et al. 2009). Only two data exceed that threshold, both of which correspond to the final engine event in flight $C_{1,57}$, which was the only time during flight that the abnormality was detected.

In contrast, as illustrated here for dataset $C_1$ and later in this section for datasets $C_2$ to $C_4$, the engine-specific approach taken by B-EVT allows novelty thresholds to be set lower than 0.5 in s$^{-1}$. This provides the ability to detect engine events with lower vibration amplitudes, as are present in each of the event datasets $C_1, \ldots, C_4$ used in this investigation.

The comparative performance of the algorithms in setting novelty thresholds for a speed bin ($b = 9$) in which no event data occur in dataset $C_1$ is shown in Figure 9.9, where both EVT methods are used with $P_e(x) = 0.99$. The novelty threshold set with B-EVT
Figure 9.8: Novelty thresholds \( x_\kappa \) set using B-EVT with \( P_e(x) = 0.99 \) for all speed bins for the 1LP tracked order, for dataset \( C_1 \). Novelty thresholds for each of the \( C_{1,1}, \ldots, C_{1,57} \) flights are shown, coloured according to flight number as indicated by the colour-bar to the right of each sub-plot. The 3 maxima in each speed bin for each “abnormal” flight \( (C_{1,49}, \ldots, C_{1,57}) \) are shown as black crosses. There are 27 crosses (9 flights \( \times \) 3 maxima per flight) for each of the 20 speed bins. (a) \( y \)-axis limited to the range \( |x| = [0, 0.5] \) in \( s^{-1} \), showing detail of the novelty thresholds; (b) enlarged \( y \)-axis showing all event data, where the engine event in flight \( C_{1,57} \) reaches \( |x| = 3 \) in \( s^{-1} \).

Figure 9.9: Novelty threshold \( x_\kappa \) set for “normal” data from speed bin \( b = 9 \) of \( N_\omega = 20 \) from dataset \( C_1 \) using different methods. For each flight, the 3 maxima from the 1IP vibration signature are shown. Novelty thresholds set using ML-EVT and B-EVT with \( P_e(x) = 0.99 \) are shown in green and black, respectively. Novelty thresholds set using (9.1) with \( K_h = 1.2 \) is shown in red; mean vibration amplitudes are shown in dashed magenta, as used in determining novelty thresholds using the heuristic method - see (9.2). Novelty thresholds for the first \( r_{\min} = 5 \) flights are shown as a thick line, during which novelty detection is not performed.
is sufficiently conservative that data from all flights lie below it, and are thus correctly classified “normal”. Novelty thresholds set with ML-EVT are initially set lower than those set with B-EVT, because ML-EVT does not take model uncertainty into account. Again, the novelty threshold set with ML-EVT (shown in green) converges towards the B-EVT novelty threshold (shown in black), yet it is low enough that it is exceeded by vibration data in “normal” flight \( C_{1,15} \), which results in a FP classification.

The novelty threshold set using the heuristic method suffers a similar problem to that of ML-EVT, where data from “normal” flights are classified FP, such as data in flight \( C_{1,14} \). After a large number of flights, the novelty threshold set using the heuristic method is approximately equal to those set using the B-EVT and ML-EVT methods.

**Dataset \( C_2 \)**

Event data for dataset \( C_2 \) are shown in Figure 9.10(a), in which novelty thresholds for varying \( P_e(x) \) are shown using B-EVT. Again, \( P_e(x) = 0.90 \) results in a novelty threshold that is too low, resulting in several FP classifications. As with dataset \( C_1 \), setting \( P_e(x) = 0.99 \) provides a good compromise between avoiding FP and FN classifications. Novelty thresholds set using \( P_e(x) = 0.999 \) are again too conservative, leading to all flights being classified as “normal”, which results in both event flights \( C_{2,23} \) and \( C_{2,27} \) being classified as FN.

Figures 9.10(b) and (c) show novelty thresholds set with B-EVT using \( P_e(x) = 0.99 \) against those obtained using the ML-EVT and heuristic methods. ML-EVT correctly classifies the event data in flights \( C_{2,23} \) and \( C_{2,27} \), but has many FP classifications of data from earlier “normal” flights. The heuristic method performs particularly poorly on this dataset, with large numbers of FP classifications.

An FP classification by the B-EVT method is shown in Figure 9.11(a), in which the use of \( P_e(x) \leq 0.99 \) results in FP classification of data observed during flight \( C_{2,6} \). This occurs because data from the first four flights are tightly clustered around 0.08 in \( \text{s}^{-1} \), while data from flight \( C_{2,6} \) occur at a much higher vibration amplitude. It may be seen from the figure that novelty thresholds compensate by increasing shortly after this flight, such that subsequent “normal” flights are classified correctly.

The same data are shown in Figures 9.11(b) and (c), which show that thresholds obtained with the ML-EVT and heuristic methods result in similar FP classifications.
Figure 9.10: Comparison of novelty threshold $x_\kappa$ set for event data from speed bin $b = 17$ of $N_\omega = 20$ of dataset $C_2$. For each flight, the 3 maxima from the 1HP vibration signature are shown. In each case, thresholds for the first $r_{\text{min}} = 5$ flights are shown by thick lines, for which novelty detection is not performed. (a) Novelty thresholds set at $P_{\text{e}}(x) = 0.90, 0.99, \text{ and } 0.999$ using B-EVT are shown in green, black, and red, respectively. (b) Novelty thresholds set at $P_{\text{e}}(x) = 0.99$ using B-EVT and ML-EVT are shown in black and green, respectively. (c) Novelty thresholds set using the heuristic method from (9.1) with $K_h = 1.2 \text{ and } 1.5$ are shown in red and green, respectively. Mean vibration amplitudes are shown in magenta, determined using (9.2).
Figure 9.11: Comparison of novelty threshold $x_\kappa$ set for “normal” data from speed bin $b = 7$ of $N_\omega = 20$ of dataset $C_2$. For each flight, the 3 maxima from the 1HP vibration signature are shown. In each case, thresholds for the first $r_{\text{min}} = 5$ flights are shown by thick lines, for which novelty detection is not performed. (a) Novelty thresholds set at $P_e(x) = 0.90, 0.99, \text{and } 0.999$ using B-EVT are shown in green, black, and red, respectively. (b) Novelty thresholds set at $P_e(x) = 0.99$ using B-EVT and ML-EVT are shown in black and green, respectively. (c) Novelty thresholds set using the heuristic method from (9.1) with $K_h = 1.2$ and 1.5 are shown in red and green, respectively. Mean vibration amplitudes are shown in magenta, determined using (9.2)
Table 9.6: Classification results for test datasets. The number of “normal” and “abnormal” flights, as labelled by engine experts, is shown for each tracked order as “Nm.” and “Ab.”, respectively. Total numbers of “normal” and “abnormal” flights are shown in the rightmost column.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total flights</th>
<th>Valid flights</th>
<th>1LP Nm.</th>
<th>1LP Ab.</th>
<th>1HP Nm.</th>
<th>1HP Ab.</th>
<th>Total Nm.</th>
<th>Total Ab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_3$</td>
<td>23</td>
<td>23</td>
<td>22</td>
<td>1</td>
<td>23</td>
<td>0</td>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td>$C_4$</td>
<td>27</td>
<td>27</td>
<td>25</td>
<td>2</td>
<td>27</td>
<td>0</td>
<td>79</td>
<td>2</td>
</tr>
<tr>
<td>$C_5$</td>
<td>87</td>
<td>82</td>
<td>82</td>
<td>0</td>
<td>82</td>
<td>0</td>
<td>246</td>
<td>0</td>
</tr>
<tr>
<td>$C_6$</td>
<td>96</td>
<td>91</td>
<td>91</td>
<td>0</td>
<td>91</td>
<td>0</td>
<td>273</td>
<td>0</td>
</tr>
<tr>
<td>$C_7$</td>
<td>72</td>
<td>67</td>
<td>67</td>
<td>0</td>
<td>67</td>
<td>0</td>
<td>201</td>
<td>0</td>
</tr>
<tr>
<td>$C_8$</td>
<td>126</td>
<td>119</td>
<td>119</td>
<td>0</td>
<td>119</td>
<td>0</td>
<td>357</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9.7: Summary of classification results for novelty detection methods on test data using metaparameters obtained from training and validation.

<table>
<thead>
<tr>
<th>B-EVT FN</th>
<th>ML-EVT FN</th>
<th>Heuristic FN</th>
<th>B-EVT FP</th>
<th>ML-EVT FP</th>
<th>Heuristic FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>39</td>
<td>141</td>
<td>1</td>
<td>530</td>
<td></td>
</tr>
</tbody>
</table>

9.5.4 Testing

Using the values of the metaparameters determined during the training and validation phases, $P_e(x) = 0.99$ for ML-EVT and B-EVT, and $K_h = 1.2$ for the heuristic method of (9.1), each method was then applied to the test datasets. The numbers of “normal” and “abnormal” classifications in each test set are shown in Table 9.6. Results obtained from testing are shown in Table 9.7.

The metaparameters selected from the training and validation data result in B-EVT correctly classifying all three of the engine events in datasets $C_3$ and $C_4$, with 39 FP classifications. This corresponds to a FP rate, as defined in (2.5), of $(39/1224) \times 100 = 3.1\%$. This is a significant improvement over the number of FP classifications obtained using ML-EVT and the heuristic method, which have FP rates of 11.5\% and 43.3\%, respectively.

Of the other approaches, only ML-EVT correctly identifies all three engine events, while the heuristic method has one FN classification. Illustrative examples are given in the following subsections.
Figure 9.12: Comparison of novelty threshold $x_n$ set for event data from speed bin $b = 17$ of $N_\omega = 20$ of dataset $C_3$ using B-EVT and other methods. For each flight, the 3 maxima from the 1LP vibration signature are shown. In each case, thresholds for the first $r_{\text{min}} = 5$ flights are shown by thick lines, for which novelty detection is not performed. (a) Novelty thresholds set at $P_e(x) = 0.90, 0.99,$ and $0.999$ using B-EVT are shown in green, black, and red, respectively. (b) Novelty thresholds set at $P_e(x) = 0.99$ using ML-EVT and B-EVT are shown in green and black, respectively. (c) Novelty thresholds set using the heuristic method from (9.1) with $K_h = 1.2$ and $1.5$ are shown in red and green, respectively. Novelty thresholds set using B-EVT at $P_e(x) = 0.99$ are shown in black. Mean vibration amplitudes are shown in magenta, as used in (9.2).

**Dataset $C_3$**

Figure 9.12(a) shows event data for the 1LP tracked order in bin $b = 17$ for dataset $C_3$. The value of the novelty threshold set using B-EVT following training and validation ($P_e(x) = 0.99$) is shown in black, with the other two novelty thresholds shown in red and green for comparison. Each novelty threshold correctly identifies the engine event that occurred during flight $C_{3,17}$, while correctly classifying all other flights as “normal”.

A comparison between novelty thresholds set with B-EVT and ML-EVT using the
values of the metaparameters set during training and validation is shown in Figure 9.12(b). As with previous examples, the ML-EVT method results in novelty thresholds that are initially too low, and which only later attain sufficiently high values such that “normal” data are correctly classified. However, thresholds set with ML-EVT correctly identify the engine event that occurred during flight $C_{3,17}$.

Novelty thresholds set using the heuristic method of (9.1) are shown in Figure 9.12(c), which successfully classify the engine event in flight $C_{3,17}$.

**Dataset $C_4$**

Figure 9.13(a) shows novelty thresholds set using B-EVT, with $P_e(x) = 0.99$, as set after training and testing, with the other two novelty thresholds shown in red and green for comparison. Novelty thresholds set at $P_e(x) \leq 0.99$ correctly classify all “normal” data, while also identifying the engine events that occurred during flights $C_{4,28}$ and $C_{4,29}$.

Comparisons between novelty thresholds set using B-EVT and ML-EVT are shown in Figure 9.13(b), using the values of the metaparameters set during training and validation. For this dataset, both methods correctly classify the “normal” flights, and the engine event that occurred during flight $C_{4,28}$.

The use of the heuristic method to set novelty thresholds for the same data is shown in Figure 9.13(c). It may be seen from the figure that using the value $K_h = 1.2$ results in a FN classification of the event in flight $C_{4,29}$. For comparison, novelty thresholds set using $K_h = 1.5$ are also shown, recalling from Table 9.5 that this value would be required in order to reduce the number of FP classifications from 155 to 68. However, the figure shows that using this value with dataset $C_4$ would result in FN classifications for both of the flights containing event data, again highlighting the inability of the current heuristic method to detect engine events without a high FP classification rate.

### 9.6 Conclusions

This chapter has described the application of novelty detection to the analysis of low-bandwidth data that summarise the vibration characteristics of an engine during a flight. The vibration signature has been shown to be a suitable representation for novelty detection methods to be used to identify in-flight events.
Figure 9.13: Comparison of novelty threshold $x_\kappa$ set for data from speed bin $b = 16$ of $N_\omega = 20$ of dataset $C_4$ using B-EVT and other methods. For each flight, the 3 maxima from the 1LP vibration signature are shown. In each case, thresholds for the first $r_{\min} = 5$ flights are shown by thick lines, for which novelty detection is not performed. (a) Novelty thresholds set at $P_e(x) = 0.90, 0.99, \text{and } 0.999$ using B-EVT are shown in green, black, and red, respectively. (b) Novelty thresholds set at $P_e(x) = 0.99$ using ML-EVT and B-EVT are shown in green and black, respectively. (c) Novelty thresholds set using the heuristic method from (9.1) with $K_h = 1.2 \text{ and } 1.5$ are shown in red and green, respectively. Novelty thresholds set using B-EVT at $P_e(x) = 0.99$ are shown in black. Mean vibration amplitudes are shown in magenta, as used in (9.2).
Motivated by the need for principled novelty detection using the low-bandwidth data from vibration signatures acquired from in-flight data, we have applied the Bayesian approach described in Chapter 8 such that uncertainty in our model of normality is quantified. We have shown that this provides robust novelty thresholds that are conservative when we have observed small numbers of flights, and which become less conservative as more flights are observed.

We have compared the Bayesian method proposed in Chapter 8 to existing heuristic methods, and to a non-Bayesian maximum-likelihood EVT method. While all EVT methods correctly identify the engine events in the test sets, the method proposed in Chapter 8 allows this to be performed while making a small number of FP classifications on normal flights. This is critical for acceptance in practice, in which alerting systems with high FP classification rates are typically ignored or deactivated (Tsien & Fackler 1997, Hann 2008).

The proposed technique allows engine-specific novelty detection, whereby novelty thresholds may be set far less conservatively than the traditional fixed limits applied by conventional engine monitoring systems. The latter are set conservatively such that they are useful for all engines of a class; by adopting an engine-specific approach, we have shown that more sensitive novelty detection is possible allowing low-amplitude vibration abnormalities corresponding to precursors of eventual engine failure to be recognised.
Chapter 10

Conclusions

10.1 Summary of Results

This thesis has described three approaches to novelty detection in jet engine vibration data:

• An approach suitable for high-bandwidth data (Chapters 5 and 6), in which features are extracted from vibration spectra, their distribution modelled during periods of “normal” operation, and which are then used to identify newly-acquired data that are “abnormal” with respect to that model.

• An approach for the analysis of very-high bandwidth data (Chapter 7), in which entire vibration spectra are modelled, rather than features extracted from them.

• An approach suited to low-bandwidth data (Chapters 8 and 9), in which engine vibration measurements observed during engine runs are summarised by a small number of data, which are then used for novelty detection.

10.1.1 Engine monitoring

In each case, the proposed methods significantly outperform the monitoring systems implemented in existing engine health monitoring systems. Typically, the latter are overly conservative, such that one configuration of the monitoring system can be applied to all engines of a particular type (so-called population-generic or fleet-generic monitoring) without suffering from large numbers of false alarms. However, this results in only engine events with very high vibration magnitude being detected, rather than subtle precursors to engine failure. The approach taken in this thesis has been to provide
engine-specific monitoring, in which the characteristics of an individual engine may be learned on-line.

In high-bandwidth monitoring, described in Chapters 5 and 6, we showed that precursors to serious engine events could be identified when they occurred several runs in advance of a final engine event. Existing engine monitoring systems, by comparison, either identified only the final engine event (in the case of two of the four datasets considered), or did not alarm at all (in the remaining two datasets).

In the very-high bandwidth monitoring approach, described in Chapter 7, we showed that, if sufficient processing resources were available, novel content in vibration spectra corresponding to subtle engine events (such as bearing-related distress) could be automatically identified, in contrast to existing EHM systems which typically did not generate an alarm.

The low-bandwidth approach described in Chapters 8 and 9 proposed a technique suitable for novelty detection constrained by the very low communication bandwidth currently available in aircraft-to-ground transmission systems. In this case, precursors to engine events were identified using the summary data, with, again, existing engine monitoring systems only able to detect the final engine event.

We note that the precursor events were labelled by engine experts retrospectively, and that the proposed techniques were able to identify such episodes correctly. The goal of the work described by this thesis has been to replicate the knowledge of engine experts in an automated system, rather than to improve upon an expert’s capability.

10.1.2 Novelty detection

In addition to applying novelty detection to jet engine vibration data, improvements in this thesis to existing novelty detection theory are also valid in the context of data-modelling problems in general.

Motivating the use of EVT

In Chapters 3 and 4, we described the limitations of existing methods of performing novelty detection, and how EVT-based techniques can be used to overcome some of the problems, such as the lack of a principled method of determining novelty thresholds given some density model, \( M(\theta) \). We discussed the disadvantages of using existing
maximum-likelihood EVT-based techniques.

**New methods for analysing multivariate data with EVT**

Chapter 5 investigated the distributions of extrema (extreme value distributions, or EVDs) from multivariate, multimodal models of normality, and showed that existing approaches do not correctly estimate such models’ EVDs. The key insight proposed in this chapter was that EVDs could be estimated by working in the (univariate) probability density space, rather than the original multivariate data space. A transform (the \( \Psi \)-transform) was proposed that maps the univariate probability density space into a domain where the EVD can be directly modelled using univariate maximum likelihood techniques in which the distribution is Gumbel. It was shown that this EVD accurately estimates the distribution of extrema for general multivariate, multimodal models, allowing the bounds of “normality” to be accurately described for such models.

The proposed multivariate EVT approach was illustrated using engine data in Chapter 6, in which it was shown to provide improved novelty detection performance when compared with existing EVT-based methods, and with existing heuristic methods.

**Performing full-spectrum novelty detection**

The analysis of full spectral data is typically avoided due to the curse of dimensionality, in which exponentially more examples of data are required to accurately model very-high dimensional data spaces. We found the EVDs corresponding to background noise in these spectra, and automatically extracted features based on their improbability with respect to these EVDs. This allows both (i) automatic feature extraction of “significant content” from spectral data, and (ii) their subsequent use for novelty detection.

We note that recent work by Hensman & Worden (2009) has tackled a similar problem, in which entire spectra were modelled using various dimensionality-reduction techniques and other multivariate analyses.

**New methods for analysis univariate data with EVT**

For estimating the EVD given small quantities of data, maximum likelihood approaches, such as those proposed in Chapter 5, can be unreliable. In such cases, it is preferable to consider the uncertainty in our estimate of the EVD, and Chapter 8 proposed a new
method for doing so based on Bayesian modelling. Existing methods require either (i) extremely large quantities of sampling, or (ii) consideration of some model parameters as being either fixed or “nuisance” parameters. The proposed method integrates out the parameters of the underlying data distribution to give an estimate of the EVD. This typically results in a “broad” EVD, which converges to the EVD estimated by maximum likelihood approaches when the number of observed data becomes sufficiently large.

10.2 Future Work

10.2.1 Validation using larger datasets

For each of the approaches to novelty detection described by this thesis, limited datasets were available due to the infrequent nature of faults in engines which are, by design, highly reliable. While the general methodology for each type of analysis has been successfully illustrated using these datasets, the values of metaparameters associated with each approach should be validated using much larger sets of data. In particular, it is not clear whether new values for the metaparameters are needed for each different type of engine, which may differ significantly in their construction. Note that this still retains an engine-specific approach, using on-line learning, but that the general metaparameters constraining each engine’s learning process could differ by general engine class.

10.2.2 From novelty detection to prognosis

Novelty detection allows the identification of “abnormal” episodes within the operation of a system, and, as described in this thesis, can be used as the basis for an alarming system. Such systems are comparable with existing engine health monitoring approaches, in which engine manufacturers need to be warned of engine deterioration as rapidly as possible. However, the ultimate goal of the monitoring of critical systems is to perform prognosis, in which the expected remaining “life” of the system is estimated. Novelty detection is the first step towards this ultimate goal, and episodes identified as “abnormal” could ultimately be used as inputs to a prognosis system. In order to achieve this, large numbers of examples of engine “abnormalities” could be used to learn which forms of abnormality led to different classes of engine events.
10.2.3 Multivariate EVT in closed form

We have provided a formulation for the closed form of an EVD for a Gaussian distribution in bivariate $Q = 2$ space. For distributions of higher dimensionality, we have relied on sampling. However, by integration of the radial EVD throughout a hypersphere of dimensionality $Q > 2$, it should be possible to obtain a formulation for the EVD for a multivariate Gaussian distribution in closed form. This would avoid the use of sampling when determining the EVD for a single Gaussian distribution of high dimensionality.

10.2.4 Multimodal EVT in closed form

The curse of dimensionality dictates that ever-increasing numbers of data are required to characterise data spaces of increasing dimensionality. However, in Chapter 5, we noted that the estimation of the EVD for a multivariate, multimodal GMM using a small number of samples (e.g., $N = 10^3$) was very close to the estimate obtained using larger quantities of samples (e.g., $N = 10^6$). This result was consistent across models of varying numbers of component distributions (including a 60-component Parzen window model), and varying model dimensionality.

This is a counter-intuitive result, which indicates that the curse of dimensionality may not be applicable when considering the tails of a model. Intuitively, one would imagine the need to generate very large numbers of data from a high-dimensional model in order to populate the (high-dimensional) tails sufficiently in order to estimate the EVD. However, our findings have indicated that this is not the case, and that smaller numbers of samples are adequate. This suggests that, sufficiently far from the modes of a model’s distribution, the probability density contours are approximately homogeneous no matter where the observer is located around the hypersphere centred near the modes of the distribution. That is, the probability contours in the tails vary approximately radially around the model, and thus could be described by a single radial EVD.

Thus, we postulate that it could be possible to model the tails of a multivariate, multimodal distribution using a simpler model, such as a multivariate, unimodal Gaussian distribution. If the closed form of the EVD for such a distribution could be found, this would allow the EVDs for multivariate, multimodal distributions to be estimated without sampling.
10.2.5 Further applications of multivariate, multimodal EVT

We have proposed methods for using EVT to determine the location of a novelty threshold corresponding to a model of normality, but the ability to characterise the tails of multivariate, multimodal distributions (Gaussian or otherwise) is particularly useful. Kernel densities are used throughout machine learning, and the capability to characterise their tails could allow the benefits of EVT to be brought to other classes of model. We have described several such methods in this thesis, such as Support Vector Machines using Gaussian kernels, neural networks using Gaussian kernels as their radial basis functions, and Hidden Markov Models using Gaussian kernels to model data pdfs in each of their constituent states.

10.2.6 Beyond EVT

While EVT provides a principled approach to describing the tails of distributions, there are numerous other fields of statistics that can be applied to the task of novelty detection. Future work could consider the use of exact value statistics, for example, in which a non-parametric approach is taken to describe the extrema generated from a model, rather than imposing the generalised extreme value distribution (GEV) or its subtypes (Gumbel, Fréchet, and Weibull) on the data. Though the latter have been shown by the Fisher-Tippett theorem to be the asymptotic attractors for a model’s EVDs for increasing $m$, the exact value approach may yield better results for small $m$.

Furthermore, EVT, which considers the distribution of the most extreme of a block of $m$ samples drawn from the underlying model, is a subset of order statistics, which considers the distribution of all $m$ samples within a block. This latter approach could be applied to novelty detection by allowing us to determine when entire blocks of data are abnormal with respect to “normal” block behaviour, rather than just characterising the behaviour of the blocks’ extrema.

10.2.7 Dynamic novelty detection

This thesis has constrained itself to static novelty detection, in which data are assumed to be independently and identically distributed ($i.i.d.$), according to some underlying distribution $D$. We estimate that distribution, $\hat{D}$, and then set novelty thresholds using
the various methods described in this thesis.

However, data observed from actual systems may deviate from i.i.d. conditions. While the techniques described in this thesis are applicable to such data, as shown by the results in earlier chapters, an approach which relaxes this assumption would be the next logical step to take. Order statistics, described above, provide one method of performing dynamic, through-time novelty detection, however the data are still limited to some block size, \( m \). Furthermore, no unique partial ordering in multivariate spaces is defined. One could order data according to their data likelihood \( p(x|\theta) \), but this would impose a generative model \( D \) on the data. Fully dynamic approaches such as Gaussian processes or probabilistic graphical models (Bishop 2006) could yield further improvements in the identification of system “abnormality” by considering the through-time behaviour of signals that are multivariate.
Appendix A

Formulation of Expectation-Maximisation

We here formulate the EM method, as introduced in Chapter 3. From (3.29), we can see that our new likelihood function \( L(\theta) = p(X, Y|\theta) \), which is a joint distribution over \( X \) and \( Y \). Our goal is to maximise the log-likelihood

\[
\ln L(\theta) = \ln p(X, Y|\theta)
\]  

(A.1)

In order to do this, we (i) first find the expectation of the log-likelihood \( \ln p(X, Y|\theta) \) with respect to our current estimate of the parameters \( \theta_{\text{old}} \) and the observed data \( X \):

\[
F(\theta, \theta_{\text{old}}) = \mathbb{E}[\ln p(X, Y|\theta) | X, \theta_{\text{old}}]\n\]

(A.2)

where we note that this \( F(\cdot) \) is just a function of variable \( \theta \) given some constants \( \theta_{\text{old}} \) and \( X \). We can expand the above by marginalising over all possible values of \( Y \),

\[
\mathbb{E}[\ln p(X, Y|\theta) | X, \theta_{\text{old}}] = \int \ln p(X, Y|\theta) p(Y|X, \theta_{\text{old}}) \, dY
\]

(A.3)

where we have introduced \( p(Y|X, \theta_{\text{old}}) \), our estimate of the distribution describing the values of the labels \( Y \), which we will obtain given our knowledge of the constants \( X \) and \( \theta_{\text{old}} \). Knowing that the labels can only take discrete values, the above becomes

\[
\int \ln p(X, Y|\theta) p(Y|X, \theta_{\text{old}}) \, dY = \sum_{k=1}^{K} \ln p(X, Y|\theta) \, P(y_k|X, \theta_{\text{old}})
\]

(A.4)

where \( P(y_k|X, \theta_{\text{old}}) \) is just the probability of the labels given the data from the previous iteration:

\[
P(y_k|X, \theta_{\text{old}}) = \frac{P(k) \, p(X|\theta_{\text{old},k})}{\sum_{j=1}^{K} P(j) \, p(X|\theta_{\text{old},j})} = P(k|X)
\]

(A.5)

\( ^1 \)Often termed the likelihood of the complete data because \( \{X, Y\} \) is the complete data set; i.e., the observed data together with their labels.
Noting also that the first term in (A.4) can be expanded (Bilmes 1998):

\[
\ln p(X, Y|\theta) = \ln \left[ p(X|Y, \theta) p(Y|\theta) \right]
\]

(A.6)

we substitute (A.5) and (A.6) into (A.4) to obtain

\[
F(\theta, \theta_{old}) = \sum_{k=1}^{K} \ln \left[ p(X|Y, \theta) p(Y|\theta) \right] P(k|X)
\]

\[
= \sum_{k=1}^{K} \left[ \ln p(X|Y, \theta) + \ln p(Y|\theta) \right] P(k|X)
\]

\[
= \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{k=1}^{K} \left[ \ln p(x|\theta_k) + \ln P(k) \right] P(k|x_i)
\]

(A.7)

noting again that this is a function of \( \theta \), where the constants \( X \) and \( \theta_{old} \) are used to determine our belief in the values of the labels \( p(y_k|X, \theta_{old}) = P(k|X) \), which is the last term in \( F(\cdot) \) above. Thus, this step (i) is the computation of \( F(\cdot) \), which is the expectation of \( \ln p(X, Y|\theta) \), and hence this is called the E-step (for “expectation”).

Step (ii) finds the new parameters \( \theta \) that maximise \( F(\cdot) \), and hence is called the M-step (for “maximisation”):

\[
\theta_{new} = \operatorname{argmax}_\theta F(\theta, \theta_{old})
\]

(A.8)

For some models, this maximisation may not be possible (Neal & Hinton 1993), and we may accept a \( \theta_{new} \) that merely increases \( F(\cdot) \). However, for a GMM, we can find a solution to (A.8) in closed form.

In order to optimise (A.8) while enforcing the constraint \( \sum_k P(k) = 1 \), we introduce a Lagrangian multiplier \( \lambda \) (Nocedal & Wright 1999) and find the value of \( \theta \) that causes the following differential to equal zero:

\[
\frac{d}{dP(k)} \left[ F + \lambda \left( \sum_{k=1}^{K} P(k) - 1 \right) \right] = 0
\]

(A.9)
giving solutions for the parameters that maximise (A.8), \( \theta_{\text{new}} = (P_{\text{new}}(k), \mu_{\text{new}}, \Sigma_{\text{new}}) \):

\[
P_{\text{new}}(k) = \frac{1}{N} \sum_{i=1}^{N} P(k|x_i)
\]

\[
\mu_{\text{new}} = \frac{\sum_{i=1}^{N} P(k|x_i)x_i}{\sum_{i=1}^{N} P(k|x_i)}
\]

\[
\Sigma_{\text{new}} = \frac{\sum_{i=1}^{N} P(k|x_i)(x_i - \mu_{\text{new}})(x_i - \mu_{\text{new}})^T}{\sum_{i=1}^{N} P(k|x_i)}
\]
Appendix B

Hyperparameters of the Conjugate Prior

We here use the functional form of the conjugation prior for data drawn from \( N(\mu, \lambda) \), with both unknown \( \mu \) and \( \lambda \), given in Bishop (2006), and extend it to formulate expressions for its four hyperparameters.

For i.i.d. data \( X \sim N(\mu, \lambda^{-1}) \), we can write the likelihood,
\[
p(X|\theta) = p(X|\mu, \lambda) = N(x_1, \ldots, x_N|\mu, \lambda) = \prod_{i=1}^{N} N(x_i|\mu, \lambda) = \prod_{i=1}^{N} \left( \frac{\lambda}{2\pi} \right)^{1/2} \exp \left\{ -\frac{\lambda}{2} (x_i - \mu)^2 \right\} = (2\pi)^{-N/2} \left[ \lambda^{1/2} \exp \left( -\frac{\lambda \mu^2}{2} \right) \right]^N \exp \left( \lambda \mu \sum_{i=1}^{N} x_i - \frac{\lambda}{2} \sum_{i=1}^{N} x_i^2 \right)
\] (B.1)

The conjugate prior must match this functional form in order to be reproducing; i.e., it must be
\[
p(\mu, \lambda) \propto \left[ \lambda^{1/2} \exp \left( -\frac{\lambda \mu^2}{2} \right) \right]^\beta \exp(A\lambda \mu - B\lambda)
\frac{\lambda^a \Gamma(a)}{\beta^b \Gamma(b)}
\]
\[
\propto \exp \left\{ \frac{-\beta\lambda}{2} \left( \mu - \frac{A}{\beta} \right) \right\} \lambda^{\beta/2} \exp \left\{ - \left( B - \frac{A^2}{2\beta} \right) \lambda \right\}
\]
\[
\propto \exp \left\{ - \frac{\beta\lambda}{2} (\mu - A')^2 \right\} \lambda^{\beta/2} \exp\{-B'\lambda\}
\]
(B.2)

which is of the form
\[
p(\mu, \lambda) = N(\mu|\mu_0, (\beta_0\lambda)^{-1}) \text{ Gam}(\lambda|a_0, b_0)
\] (B.3)
where $\mu_0 = A'$, $\beta_0 = \beta$, $a_0 = 1 + (\beta/2)$, and $b_0 = B'$. Thus, the posterior obtained after observing $N$ data will take this form, and have hyperparameters $\{\mu_N, \beta_N, a_N, b_N\}$, for which we require expressions.

Comparing terms on both sides of the Bayesian equation (8.10), recalling that the posterior is Normal-gamma, the data likelihood is Normal, and the prior is Normal-gamma:

$$N(\mu|\mu_N, (\beta_N \lambda)^{-1}) \text{Gam}(\lambda|a_N, b_N) =$$

$$K \prod_{i=1}^{N} N(x_i|\mu, \lambda) N(\mu|\mu_0, (\beta_0 \lambda)^{-1}) \text{Gam}(\lambda|a_0, b_0) \quad (B.4)$$

where $K$ is a constant of proportionality, being the denominator in the Bayesian equation, $p(X) = \int p(X|\theta) p(\theta) \, d\theta$, which we will not need to evaluate.

Expanding the left-hand side of (B.4) gives,

$$N(\mu|\mu_N, (\beta_N \lambda)^{-1}) \text{Gam}(\lambda|a_N, b_N)$$

$$= \left(\frac{\beta_N \lambda}{2\pi}\right)^{1/2} \exp \left\{ -\frac{\beta_N \lambda}{2} (\mu - \mu_N)^2 \right\} \frac{b_N^{a_N}}{\Gamma(a_N)} \lambda^{a_N - 1} \exp\{-b_N \lambda\}$$

$$= \left(\frac{\beta_N^{1/2} b_N^{a_N}}{(2\pi)^{1/2} \Gamma(a_N)}\right) \lambda^{a_N - 1/2} \exp \left\{ -\frac{\beta_N \lambda}{2} (\mu - \mu_N)^2 - b_N \lambda \right\} \quad (B.5)$$

$$= K' \lambda^{a_N - 1/2} \exp \left\{ -\frac{\beta_N \lambda}{2} (\mu - \mu_N)^2 - b_N \lambda \right\}$$
while expanding the right-hand side of (B.4) gives,

\[
K \prod_{i=1}^{N} \left[ N(x_i | \mu, \lambda) \right] \frac{\lambda^N}{2^N} \text{exp} \left\{ -\frac{\lambda}{2} (x_i - \mu)^2 \right\} \left( \frac{\beta_0 \lambda}{2} \right)^{1/2} \\
\exp \left\{ -\frac{\beta_0 \lambda}{2} (\mu - \mu_0)^2 \right\} b_0^{\lambda a_0 - 1} \Gamma(a_0) \exp \{-b_0 \lambda\} = K'' \lambda^{N/2} \exp \left\{ -\frac{N \lambda \mu^2}{2} + \lambda \mu \sum_{i=1}^{N} x_i - \frac{\lambda}{2} \sum_{i=1}^{N} x_i^2 - \frac{\beta_0 \lambda}{2} (\mu - \mu_0)^2 - b_0 \lambda \right\} \tag{B.6}
\]

Comparing the powers of \( \lambda \) in (B.5) and (B.6) gives \( a_N \),

\[
a_N = a_0 + \frac{N}{2} \tag{B.7}
\]

while comparing the exponents of (B.5) and (B.6),

\[
-\frac{\beta N \lambda}{2} (\mu - \mu_N)^2 - b_N \lambda = -\frac{N \lambda \mu^2}{2} + \lambda \mu \sum_{i=1}^{N} x_i - \frac{\lambda}{2} \sum_{i=1}^{N} x_i^2 - \frac{\beta_0 \lambda}{2} (\mu - \mu_0)^2 - b_0 \lambda \tag{B.8}
\]

and collecting terms on both sides,

\[
\left( -\frac{\beta N}{2} \right) \lambda \mu^2 + (\beta N \mu_N) \lambda \mu - \left( \frac{\beta N \mu_N^2}{2} + b_N \right) \lambda = \left( -\frac{N + \beta_0}{2} \right) \lambda \mu^2 + \left( \sum_{i=1}^{N} x_i + \beta_0 \mu_0 \right) \lambda \mu - \left( \frac{\beta_0 \mu_0^2}{2} + \sum_{i=1}^{N} x_i^2 + b_0 \right) \lambda \tag{B.9}
\]

gives \( \beta_N \) by comparing the terms in \( \mu \lambda^2 \) from (B.9),

\[
-\frac{\beta_N}{2} = -\frac{N + \beta_0}{2} \tag{B.10}
\]

and gives \( \mu_N \) by comparing the terms in \( \mu \lambda \) from (B.9),

\[
\beta_N \mu_N = \sum_{i=1}^{N} x_i + \beta_0 \mu_0 \tag{B.11}
\]

\[
\mu_N = \frac{\beta_0 \mu_0 + \sum_{i=1}^{N} x_i}{\beta_0 + N}
\]
To find $b_N$, we compare the terms in $\lambda$ from (B.9),

\[
\frac{\beta_N \mu_N^2}{2} + b_N = \frac{\beta_0 \mu_0^2}{2} + \sum_{i=1}^{N} \frac{x_i^2}{2} + b_0
\]

\[
b_N = b_0 + \frac{1}{2} \left[ \beta_0 \mu_0^2 + \sum_{i=1}^{N} x_i^2 - \beta_N \mu_N^2 \right]
\]

\[
= b_0 + \frac{1}{2} \left[ \sum_{i=1}^{N} x_i^2 + \beta_0 \mu_0^2 - \left( \frac{\beta_0 \mu_0 + \sum_{i=1}^{N} x_i}{\beta_0 + N} \right)^2 \right]
\]

\[
= b_0 + \frac{1}{2} \left[ \sum_{i=1}^{N} (x_i - \mu_{ML})^2 + \frac{\beta_0 N}{\beta_0 + N} (\mu_{ML} - \mu_0)^2 \right]
\]

where we have used the identity $\mu_{ML} = \frac{1}{N} \sum_{i=1}^{N} x_i$. Note that we can write (B.12) in the form

\[
b_N = b_0 + \frac{1}{2} \left[ \sum_{i=1}^{N} x_i^2 - \frac{1}{N} \left( \sum_{i=1}^{N} x_i \right)^2 + \frac{\beta_0 N}{\beta_0 + N} \left( \frac{1}{N} \sum_{i=1}^{N} x_i - \mu_0 \right)^2 \right]
\]

in which only sums and sums-of-squares are used, rather than the sample mean, $\mu_{ML}$. This avoids the requirement to store the whole dataset $X$, making it the more suitable form for on-line monitoring, in which computational resources may be limited.
Appendix C

Sampling the Posterior $p(\mu, \lambda)$

We here derive the normalising constant $Q$ for sampling the posterior distribution $p(\mu, \lambda)$, described in Chapter 8.3.7.

From (8.21), it can be seen that we can obtain an expression for the normalising constant $Q$ by expanding $b_N^{a_N}$ into the form $[Qc_N^{a_N}]$. From our expression for $a_N$ (B.7), $a_N \in \mathbb{Z}$ for even $N$, and $a_N \in \{\mathbb{Z} + \frac{1}{2}\}$ for odd $N$.

To find $Q$ for even $N$,

$$b_N^{a_N} = b_N^{a_N} \sum_{k=0}^{a_N} \binom{a_N}{k} b_N^{k-a_N} \gamma^{a_N-k} = \sum_{k=0}^{a_N} \binom{a_N}{k} b_N^{k-a_N} \gamma^{a_N-k} = \frac{\sum_{k=0}^{a_N} \binom{a_N}{k} b_N^{k-a_N} \gamma^{a_N-k}}{b_N + \gamma}$$  \hspace{1cm} (C.1)

for dummy variable $\gamma$, and where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient. Substituting $\gamma = \frac{\beta_N}{2} (K - \mu)^2$ in (C.1) gives

$$b_N^{a_N} = \frac{(b_N + \frac{\beta_N}{2} (K - \mu)^2)^{a_N}}{\sum_{k=0}^{a_N} \binom{a_N}{k} b_N^{k-a_N} \left\{ \frac{\beta_N}{2} (K - \mu)^2 \right\}^{a_N-k}}$$  \hspace{1cm} (C.2)

Thus for even $N$, normalising constant $Q$ in (8.21) is the denominator of (C.2),

$$Q^{-1}_{even} = \sum_{k=0}^{a_N} \binom{a_N}{k} b_N^{k-a_N} \left\{ \frac{\beta_N}{2} (K - \mu)^2 \right\}^{a_N-k}$$  \hspace{1cm} (C.3)

Performing similar steps for odd $N$, we find $Q$,

$$Q^{-1}_{odd} = \sqrt{c_N} \sum_{k=0}^{a_N-1/2} \binom{a_N - 1/2}{k} b_N^{k-a_N} \left\{ \frac{\beta_N}{2} (K - \mu)^2 \right\}^{a_N-k-1/2}$$  \hspace{1cm} (C.4)

ARINC (2005), ‘758-2 standard for a Communications Management Unit (CMU) Mark 2’.


Box, G. & Tiao, G. (1973), *Bayesian Inference in Statistical Analysis*, Addison-Wesley, Reading, Massachusetts, USA.


226


HereWatch (2003), ‘Aerosupport HercWatch product specification’.


Larsen, J. (2003), Distribution of the density of a gaussian mixture, Technical report, Department of Informatics and Mathematical Modelling, Technical University of Denmark.


Tukey, J. (1977), Exploratory Data Analysis, Addison-Wesley, Reading, Massachusetts, USA.