Methods for Evaluating and Improving Mortality Prediction in Intensive Care

Nic Dunkley

Department of Engineering
University of Oxford

Supervised by
Prof. Gari D. Clifford
Dr. Andrew A. Kramer
Prof. David A. Clifton

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of

Doctor of Philosophy

Exeter College December 2017
Abstract

Methods for Evaluating and Improving Mortality Prediction in Intensive Care

Nicholas Dunkley
Exeter College
Department of Engineering Science

Doctor of Philosophy (D.Phil)
University of Oxford
Hilary Term, 2017

The sustained spread of innovation, increased accountability and better focusing of resources are key to improving Intensive Care Unit (ICU) efficiency and efficacy. Mortality prediction models (MPMs), which estimate patients’ risk of mortality as a function of their clinical profiles, could contribute greatly to this cause through their use in ICU benchmarking and patient deterioration alerting (risking). However, current MPMs have failed to achieve widespread clinical acceptance because the processes used in their construction, herein referred to as mortality prediction model constructors (MPMCs), do not focus sufficiently on either benchmarking or alerting, for the reasons to be expounded and explored. The main MPMCs of the MPM literature are reviewed and their common approaches identified and combined to form the ‘Standard MPMC’. For each of the MPMs and performance estimates output by both the Standard MPMC and its potential alternatives, their applicability to both benchmarking and risking is examined both empirically and though simulation. First, the Standard MPMC is adjusted to output appropriate benchmarking performance estimates. This includes a demonstration that alternatives to the Area Under the Curve (AUC) and Hosmer-Lemshow Statistic (H-L) can better discriminate between MPMs for use in ICU benchmarking; with the proportion of times that the different metrics select the wrong MPM found to be 0.38, 0.47 and 0.12, for the AUC, H-L and best alternative, respectively. Second, methods for improving the risking performance of MPMs are developed and evaluated. Such investigations include a quantification of the differences in the risking performance of MPMs expected from using each of: alternative machine learning techniques (AUC from 0.845 to 0.860), moving from small homogeneous to large non-homogeneous ICU Databases (AUC from 0.868 to 0.876), and increasing the quantity of clinical information per patient (AUC from 0.876 to 0.890). Third, methods are explored for improving the accuracy of, and providing confidence intervals for, the performance estimates output by MPMCs.
Acknowledgements

I would like to thank Cerner Corporation for providing the funding, data and inspiration that made this research possible. Many thanks also go to my supervisors, Prof. Gari Clifford, Dr. Andrew Kramer and Prof. David Clifton, for their years of support, guidance and advice.
Table of contents

List of figures x

List of tables xiii

1 Introduction 1
   1.1 Mortality Prediction Model Constructors (MPMCs) . . . . . . . . . . . . . 3
   1.2 The Standard MPMC . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   1.3 Issues with the Standard MPMC . . . . . . . . . . . . . . . . . . . . . . 7
      1.3.1 Generic Issues . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
      1.3.2 Risking-Specific Issues . . . . . . . . . . . . . . . . . . . . . . 8
      1.3.3 Conflation of Risking and Benchmarking Performance . . . . . . . 9
      1.3.4 Assuming Complete Case-mix Adjustment . . . . . . . . . . . . . 11
   1.4 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
   1.5 Overview of thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2 Background 14
   2.1 The APACHE IV MPMC . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
      2.1.1 Specification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
      2.1.2 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      2.1.3 Adaptation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
   2.2 Literature Review . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>MPM Construction</td>
<td>22</td>
</tr>
<tr>
<td>2.2.2</td>
<td>MPM Comparison</td>
<td>26</td>
</tr>
<tr>
<td>2.2.3</td>
<td>MPM Validation</td>
<td>28</td>
</tr>
<tr>
<td>2.2.4</td>
<td>MPM Review</td>
<td>29</td>
</tr>
<tr>
<td>2.3</td>
<td>Machine Learning in Critical Care</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>Data Preparation</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Events and ICU Databases</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Patient Profiles</td>
<td>37</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Prediction Time Exclusions</td>
<td>37</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Transfer Event Obfuscation</td>
<td>38</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Treatment Event Retention</td>
<td>39</td>
</tr>
<tr>
<td>3.3</td>
<td>Response Selection</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Feature Extraction</td>
<td>46</td>
</tr>
<tr>
<td>3.4.1</td>
<td>The Acute Physiology Score</td>
<td>47</td>
</tr>
<tr>
<td>3.4.2</td>
<td>The Standard Extractor</td>
<td>49</td>
</tr>
<tr>
<td>3.4.3</td>
<td>The Extended Extractor Inducer</td>
<td>51</td>
</tr>
<tr>
<td>3.5</td>
<td>Stay Exclusions</td>
<td>51</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Unconditional Exclusions</td>
<td>52</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Conditional Exclusions</td>
<td>52</td>
</tr>
<tr>
<td>3.6</td>
<td>Preprocessing</td>
<td>53</td>
</tr>
<tr>
<td>3.7</td>
<td>The APACHE Database</td>
<td>53</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Feature Distributions</td>
<td>54</td>
</tr>
<tr>
<td>3.8</td>
<td>The MIMIC II Database</td>
<td>58</td>
</tr>
<tr>
<td>4</td>
<td>Machine Learning</td>
<td>59</td>
</tr>
<tr>
<td>4.1</td>
<td>The Null Inducer (NULL)</td>
<td>62</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2</td>
<td>Nearest-Neighbors (NN)</td>
<td>63</td>
</tr>
<tr>
<td>4.3</td>
<td>Gaussian Classifiers (GC)</td>
<td>66</td>
</tr>
<tr>
<td>4.4</td>
<td>Regularized Logistic Regression (RLR)</td>
<td>67</td>
</tr>
<tr>
<td>4.5</td>
<td>The APACHE IV Inducer (APACH)</td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Classification and Regression Trees (CART)</td>
<td>73</td>
</tr>
<tr>
<td>4.7</td>
<td>Random Forests (RF)</td>
<td>75</td>
</tr>
<tr>
<td>4.8</td>
<td>Bayesian Additive Regression Trees (BART)</td>
<td>76</td>
</tr>
<tr>
<td>4.9</td>
<td>The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>4.9.1  Forest Structure</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>4.9.2  Initialization</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>4.9.3  Updating</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>4.9.4  Making Predictions</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>Inference</td>
<td>87</td>
</tr>
<tr>
<td>5.1</td>
<td>Mortality Prediction Model Inducers (MPMIs)</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>Partitioning Matrix Inducers</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>5.2.1  Holdout</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>5.2.2  Cross-Validation</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>5.2.3  Repeat Cross-Validation</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>5.2.4  Splitting on Patients vs ICUs</td>
<td>95</td>
</tr>
<tr>
<td>5.3</td>
<td>The Risk Array Inducer</td>
<td>96</td>
</tr>
<tr>
<td>5.4</td>
<td>Mortality Prediction Model Evaluators</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>5.4.1  The Standardized Log-Likelihood</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>5.4.2  The $\chi^2$ Statistic</td>
<td>99</td>
</tr>
<tr>
<td>5.5</td>
<td>Performance Point Estimate Inducers</td>
<td>99</td>
</tr>
<tr>
<td>5.6</td>
<td>Standard Error Estimate Inducers</td>
<td>100</td>
</tr>
<tr>
<td>5.7</td>
<td>Hyper-Parameter Selection</td>
<td>103</td>
</tr>
</tbody>
</table>
### Table of contents

5.8 MPMC Specification and Application ............................................. 106

6 Simulations .................................................................................. 110

6.1 The Simulator ............................................................................ 112
  6.1.1 Patients and ICUs .............................................................. 113
  6.1.2 MPMIs ............................................................................ 117
  6.1.3 Risk Array Generation ...................................................... 120

6.2 The Representative MPMC ......................................................... 120
  6.2.1 Specification .................................................................... 121
  6.2.2 Results and discussion ...................................................... 124

6.3 Risk Array Statistics ................................................................. 126
  6.3.1 Specification .................................................................... 127
  6.3.2 Evaluation ....................................................................... 129
  6.3.3 Results and discussion ...................................................... 130

6.4 Parameter Fitting ....................................................................... 131
  6.4.1 Specification .................................................................... 131
  6.4.2 Methods .......................................................................... 135
  6.4.3 Results and discussion ...................................................... 137

7 Performance Evaluators ............................................................... 140

7.1 ICU Evaluators ......................................................................... 141
  7.1.1 The Standardized Mortality Ratio (SMR) .......................... 141
  7.1.2 Issues with the SMR ......................................................... 143
  7.1.3 The Posterior Model ......................................................... 146

7.2 Mortality Prediction Model Evaluators ....................................... 147
  7.2.1 The Standard MPM Evaluators ....................................... 148
  7.2.2 Issues with the Standard MPM Evaluators ...................... 149
### Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.3 Membership-Dependent MPM Evaluators</td>
<td>151</td>
</tr>
<tr>
<td>7.2.4 Further Membership-Independent MPM Evaluators</td>
<td>152</td>
</tr>
<tr>
<td>7.3 Assessing the Performance of Performance Evaluators</td>
<td>154</td>
</tr>
<tr>
<td>7.3.1 Specification</td>
<td>154</td>
</tr>
<tr>
<td>7.3.2 Methods</td>
<td>158</td>
</tr>
<tr>
<td>7.3.3 Results and discussion</td>
<td>158</td>
</tr>
<tr>
<td>8 Partitioning and the Assessment of Generalization Error</td>
<td>165</td>
</tr>
<tr>
<td>8.1 Issues with Patient Holdout</td>
<td>166</td>
</tr>
<tr>
<td>8.2 Measuring Party Performance</td>
<td>169</td>
</tr>
<tr>
<td>8.3 Defining Party Performance</td>
<td>171</td>
</tr>
<tr>
<td>8.4 Evaluating Party Performance</td>
<td>175</td>
</tr>
<tr>
<td>8.5 Results</td>
<td>176</td>
</tr>
<tr>
<td>8.6 Discussion</td>
<td>186</td>
</tr>
<tr>
<td>8.7 Conclusions</td>
<td>189</td>
</tr>
<tr>
<td>9 Improving Internal Risking</td>
<td>191</td>
</tr>
<tr>
<td>9.1 Potential Improvements Under Consideration</td>
<td>192</td>
</tr>
<tr>
<td>9.1.1 Alternative Machine Learning Techniques</td>
<td>193</td>
</tr>
<tr>
<td>9.1.2 Increased Patient Homogeneity</td>
<td>193</td>
</tr>
<tr>
<td>9.1.3 More Prolific Feature Extractor Inducers</td>
<td>194</td>
</tr>
<tr>
<td>9.1.4 Alternative Prediction Times</td>
<td>194</td>
</tr>
<tr>
<td>9.2 Assessing Performance Improvements</td>
<td>194</td>
</tr>
<tr>
<td>9.2.1 Option Set Definitions</td>
<td>195</td>
</tr>
<tr>
<td>9.2.2 Option Set Selection</td>
<td>198</td>
</tr>
<tr>
<td>9.3 Evaluation Methods</td>
<td>201</td>
</tr>
<tr>
<td>9.4 Results and Discussion</td>
<td>202</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>9.4.1</td>
<td>Machine Learning Techniques</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Patient Homogeneity</td>
</tr>
<tr>
<td>9.4.3</td>
<td>Feature Extractor Inducers</td>
</tr>
<tr>
<td>9.4.4</td>
<td>Prediction Times</td>
</tr>
<tr>
<td>9.5</td>
<td>Conclusions</td>
</tr>
<tr>
<td>10</td>
<td>Conclusions and Further Work</td>
</tr>
<tr>
<td></td>
<td>References</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Glossary of Terms</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Glossary of Symbols</td>
</tr>
<tr>
<td>Appendix C</td>
<td>The Standard Preprocessor Inducer</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Further Normal Probability Plots</td>
</tr>
<tr>
<td>Appendix E</td>
<td>Risking vs Benchmarking Feature Importance</td>
</tr>
</tbody>
</table>
## List of figures

1.1 The relationship between age and mortality rate .................................. 10
3.1 The relationship between ICU size and mortality rate ............................... 55
3.2 Temperature, white blood cell count and Glasgow Coma Scale distributions 56
4.1 Age and GCS of patients in example training sets ................................. 62
4.2 Example risk estimates output by Nearest Neighbors ............................. 65
4.3 Example risk estimates output by Gaussian Classifiers ........................... 68
4.4 Example risk estimates output by logistic regression models .................. 71
4.5 Example risk estimates output by a classification tree ......................... 75
4.6 Example risk estimates output by a Random Forest .............................. 77
4.7 Example risk estimates output by Bayesian Additive Regression Trees ..... 78
4.8 Example splits used by the Bayesian Ensemble of Additive Sigmoidal Trees (BEAST) ............................................................... 80
4.9 Example trees used by the BEAST ...................................................... 81
4.10 Example risk estimates output by a single BEAST forest ..................... 82
4.11 Example risk estimates output by the BEAST ..................................... 86
5.1 Diagrammatic representation of an MPM ............................................. 88
5.2 Diagrammatic representation of an $MPM_i^0$ ..................................... 90
5.3 Diagrammatic representation of the Risk Array Inducer ....................... 98
List of figures

5.4 Diagrammatic representation of the Hyper-Parameter Selector .................. 105
5.5 Diagrammatic representation of an MPMC ................................. 109
6.1 Performance of the MPMI's in the Representative MPMC .................... 124
7.1 Effects of different priors on quality of care p-values in the presence of
    minimal evidence ........................................................................ 144
7.2 Effects of different priors on quality of care p-values in the presence of
    substantial evidence .................................................................... 145
8.1 Distribution of SLL performance point estimate errors for different parties
    applied to simulated data ............................................................... 178
8.2 Distribution of $\chi^2$ performance point estimate errors for different parties
    applied to simulated data ............................................................... 179
8.3 Distribution of SLL performance point estimate errors for different parties
    applied to the APACHE Database .................................................... 180
8.4 Distribution of SLL standard error estimates for different parties applied to
    simulated data ............................................................................. 183
8.5 Distribution of $\chi^2$ standard error estimates for different parties applied to
    simulated data ............................................................................. 184
8.6 Distribution of SLL standard error estimates for different parties applied to
    the APACHE Database ................................................................. 185
9.1 Effects of different learner inducers on risking performance .................... 203
9.2 Effects of different option sets on risking performance ............................ 204
D.1 Mean arterial blood pressure, heart rate and respiratory rate distributions .. 241
D.2 Urine output, hematocrit and blood sodium level distributions .................. 242
D.3 Blood urea nitrogen levels, creatinine and blood glucose level distributions . 243
D.4  Albumin, bilirubin and $PaO_2/FiO_2$ distributions  . . . . . . . . . . . . . . 244
D.5  Partial pressure of oxygen in the blood, partial pressure of carbon dioxide in
      the blood and blood pH distributions  . . . . . . . . . . . . . . . . . . . . 245
D.6  Age and previous length of stay distributions  . . . . . . . . . . . . . . . 246
E.1  Benchmarking versus risking importance of the Standard Features  . . . 249
List of tables

2.1 Summary of the most commonly used MPMs ........................................ 23

3.1 Example ICU Database events ............................................................ 36

3.2 Example patient profile ................................................................. 41

3.3 Effect of heart rate on the Acute Physiology Score .............................. 48

3.4 List of Standard Features ............................................................... 50

4.1 List of Learner Inducers ................................................................. 60

5.1 Table of $MPMI^0$s ................................................................. 89

5.2 List of $MPMI^1$s ................................................................. 91

5.3 Partitioning matrix inducer properties .............................................. 93

5.4 Summary of MPMC options ......................................................... 107

5.5 Standard MPMC specification ....................................................... 108

6.1 Representative MPMC specification ................................................. 122

6.2 Risk array statistic values ............................................................. 131

6.3 Simulation parameter values .......................................................... 137

7.1 MPM evaluator discrimination using $k^{4P}$ ...................................... 159

7.2 MPM evaluator discrimination ........................................................ 160

7.3 Performance of MPMs selected by different MPM evaluators using $k^{4P}$ 161
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.4</td>
<td>Performance of MPMs selected by different MPM evaluators using $k^{19P}$</td>
<td>162</td>
</tr>
<tr>
<td>7.5</td>
<td>Performance of MPMs selected by different MPM evaluators using $k^{AB}$</td>
<td>163</td>
</tr>
<tr>
<td>8.1</td>
<td>Table of parties</td>
<td>167</td>
</tr>
<tr>
<td>8.2</td>
<td>Specification for MPMC used to compare difference partitioning approaches</td>
<td>177</td>
</tr>
<tr>
<td>8.3</td>
<td>Further aspects of party performance</td>
<td>182</td>
</tr>
<tr>
<td>9.1</td>
<td>Risking option sets</td>
<td>199</td>
</tr>
<tr>
<td>9.2</td>
<td>Performance assessment MPMC specification</td>
<td>201</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

An intensive care unit (ICU) is a hospital department that treats patients who are in a critically ill or unstable condition. In 2002 a systematic review found that the mortality rate in ICUs with high-intensity staffing was substantially lower than in ICUs with low-intensity staffing (mortality ratio of 0.61 with a 95% confidence interval of [0.50-0.75]) [67]. This indicates that significantly increasing ICU resources, or alternatively improving ICU efficiency while maintaining current resource levels, could offer substantial reductions in ICU mortality rates. With approximately 500,000 deaths occurring annually in ICUs in the United States [86] this would equate to a substantial reduction in overall mortality rates. Given the high costs of these ICUs (during 2005 it was found that the annual cost of ICUs in the United States was $81 billion [31]), ever increasing ICU populations [81] and lack of specialists [61], it seems likely that substantially increasing ICU staffing per patient would be infeasible, which leaves improvements to efficiency as perhaps the most viable way to reduce ICU mortality rates.

The widespread implementation of accurate mortality prediction models (MPMs) is one promising method for providing such ICU efficiency improvements. An MPM is a method for evaluating a patient’s ‘risk’; the probability that the patient will die within a specific time frame, based on his or her clinical profile. The risk can be used as a surrogate for a
patient’s overall health, which may then be used for either ‘patient specific decision support’ or ‘benchmarking’.

In ‘patient specific decision support’ the risk is used to help inform treatment decisions. Patient deterioration alerting is an example of patient specific decision support in which physicians are alerted to patients who have undergone a recent and substantial increase in risk, potentially reducing response times and subsequently improving recovery chances. For brevity, patient specific decision support shall be referred to as ‘risking’ for the remainder of this thesis.

In ‘benchmarking’ the risk is used to help assess the relative performance of different treatments or institutions (such as ICUs) so as to help identify and spread best practice through the critical care community. The risk is used to evaluate the expected number of deaths within a given treatment group or institution. This value is then compared against the actual number of deaths, with a significantly lower actual number of deaths taken as evidence of superior treatment or institutional performance. In this thesis the consideration of benchmarking is limited to the benchmarking of ICUs.

The focus of this thesis is on the methods used to construct MPMs, henceforth referred to as MPM Constructors (MPMCs), and how they might be altered so as to improve both benchmarking and risking. The primary approach is to examine the current MPMCs, identify their potential weaknesses, and propose and quantitatively evaluate viable alternatives. The secondary approach is to examine the effects on MPM performance of applying the same MPMCs to different ICU Databases, with the aim of helping to improve future ICU Database design so as to facilitate the construction of superior MPMs.

Notably the patient profiles to which the MPMs of this thesis are applied do not contain any high frequency time series. The highest sampling rates included here are of approximately one recording per 15 minutes, and tend to be those reported by nurses. This is in stark contrast
to the high frequency measurements reported by autonomous bedside monitors, from which continuous streams of numerous values per second can be retrieved.

The remainder of this Chapter is organized as follows: First the desirable properties of an MPMC’s outputs are discussed (Section 1.1). Next the ‘Standard MPMC’ is introduced as an embodiment of the MPMCs that are most widespread within the ICU mortality prediction literature (Section 1.2). Potential issues with the Standard MPMC are then examined and potential solutions briefly discussed (Section 1.3). Finally the contributions (Section 1.4) and structure (Section 1.5) of this thesis are summarized.

1.1 Mortality Prediction Model Constructors (MPMCs)

A patient’s ‘risk’ gives the probability that the patient will die within a specific time frame, such as between ICU admission and hospital discharge. A mortality prediction model (MPM) is a method for estimating a patient’s risk as a function of their clinical profile. An ICU Database is a collection of patient medical records. A mortality prediction model constructor (MPMC) is taken to be a method that, when applied to an ICU Database, generates at least one MPM in order to provide one or more of the following outputs:

- An MPM.

- A generalization error estimate for each of two or more MPMs; where the ‘generalization error’ of an MPM is a measure of the performance that an MPM is expected to exhibit when applied to new patients.

- A ‘quality of care’ estimate for each ICU in the ICU Database to which the MPMC was applied; where the quality of care is a measure of the extent to which a given ICU reduces an ‘average’ ICU patient’s risk.

Suppose that an MPM has been output by applying an MPMC to an ICU Database containing patients from a given set of ICUs. Then such an MPM shall be referred to as an
‘internal’ MPM if it is intended for use on new patients from within this set of ICUs or as an ‘external’ MPM if it is intended for use on new patients from outside of this set of ICUs. Suppose further that a patient’s risk can be divided into two parts: a ‘patient component’ that specifies how unwell a given patient is and a ‘quality component’ that indicates the quality of care provided by the ICU in which the patient is treated. Then an internal MPM may use information regarding the quality of care provided by the ICU treating the patient for which a risk estimate is to be made, while an external MPM cannot, such that an internal MPM may account for both the patient component and the quality component while an external MPM may only account for the former.

In this thesis an ICU’s case-mix is taken to be a measure of the extent to which an average patient admitted to the ICU is more unwell than an average patient admitted to an average ICU. In benchmarking the aim is for an MPM to account for the differences in the case-mix of ICUs, such that any remaining differences in predicted and actual mortality rates can be attributed to differences in ICU quality of care. Allowing a benchmarking MPM to account for ICU quality of care would invalidate this procedure and as a result there would be no reason for an MPMC to output an MPM tailored to internal benchmarking. Instead the primary output of an MPMC for use in internal benchmarking should be a set of quality of care estimates; one for each ICU in the ICU Database to which the MPMC is applied.

MPMCs that output a single MPM should also output an associated generalization error estimate to inform the potential user of the limits of the output MPM. To know how reliable an output generalization error estimate is it is also important to output an estimate for the degree of confidence associated with the generalization error estimate. In this thesis the generalization error estimate and its associated degree of confidence shall be expressed as a ‘performance point estimate’ and a ‘standard error estimate’ which provide the mean and standard deviation associated with the generalization error, respectively. Performance point estimates and standard error estimates may be classed as either risking or benchmarking
1.1 Mortality Prediction Model Constructors (MPMCs)

depending on which aspect of MPM performance they measure. They may also be classed as either internal or external depending on which patients are assumed to be the target of their associated MPM.

It is demonstrated in Section 6.4.3 that the contribution to a patient’s risk from the patient component far exceeds that from the quality component. Given that one can only use the quality of care component to inform risk estimates for internal risking MPMs, and that doing so only offers modest increases in accuracy, it is convenient to restrict internal risking MPMs to only account for the patient component. This then allows internal risking MPMs and external risking MPMs to be conflated into ‘risking MPMs’ and allows all of the MPMs considered in this thesis to be trained without the use of information pertaining to the care component. Similarly internal risking point error estimates and external risking point error estimates can be conflated into ‘risking point error estimates’. However, the same cannot be achieved for internal risking standard error estimates and external risking standard error estimates as the latter will tend to be much larger in situations where there are a limited number of ICUs in the ICU Database to which the MPMC is applied. Given these considerations an MPMC should ideally provide at least one of the following sets of ‘preferred output groups’:

1. Internal Risking: A risking MPM with a risking point error estimate and an associated internal risking standard error estimate.

2. External Risking: A risking MPM with a risking point error estimate and an associated external risking standard error estimate.

3. External Benchmarking: A benchmarking MPM with a benchmarking point error estimate and an associated external benchmarking standard error estimate.
4. Internal Benchmarking: A quality of care estimate for each ICU contained within the ICU Database to which the MPMC is applied, each with an associated confidence interval.

5. Internal Risking Comparison: Multiple risking point error estimates with associated internal risking standard error estimates and pairwise evaluations of the statistical significance associated with the difference between the risking point error estimates.

6. External Risking Comparison: Multiple risking point error estimates with associated external risking standard error estimates and pairwise evaluations of the statistical significance associated with the difference between the risking point error estimates.

7. External Benchmarking Comparison: Multiple benchmarking point error estimates with associated external benchmarking standard error estimates and pairwise evaluations of the statistical significance associated with the difference between the benchmarking point error estimates.

1.2 The Standard MPMC

Here the ‘Standard MPMC’ is introduced as an embodiment of the most common MPMCs from the MPM literature (detailed in Chapter 2) and it is intended to act as a baseline against which alternative MPMCs can be compared. The Standard MPMC outputs an MPM which predicts how likely an ICU patient is to die before hospital discharge, using information recorded during the first 24 hours of the patient’s ICU stay. A summary of the steps applied by the Standard MPMC is as follows:

1. An ICU Database containing many patients (~100,000) from multiple ICUs (~100) is selected.

2. Patients meeting various exclusion criteria are removed.
3. The ‘prediction time’ (the final time at which new information can be used to assess a patient’s risk) is set to be between 4 and 24 hours, depending on a patient’s ICU length of stay.

4. Patients are randomly partitioned into a ‘training set’ containing 60% of patients and a ‘test set’ containing 40% of patients, without regard to how patients are distributed among ICUs.

5. A small number (~30) of clinically relevant features are selected heuristically and extracted.

6. Features are pre-processed by removing outliers and missing values replaced through median imputation.

7. An MPM is trained by applying regularized logistic regression with square terms to the patients in the training set.

8. The MPM is used to make predictions on the test set, and the Area Under the Curve (AUC) and Hosmer-Lemshow Statistic (H-L) are evaluated on these predictions in order to generate performance point estimates.

9. The MPM and the performance point estimates are output.

Motivation for the exact form of the steps included in the Standard MPMC is delayed until Chapter 2, after the most common MPMCs of the MPM literature have been introduced.

1.3 Issues with the Standard MPMC

That the Standard MPMC does not provide any of the seven preferred output groups enumerated in Section 1.1 indicates that it does not excel at any particular task. This then motivates
1.3 Issues with the Standard MPMC

the focus of this thesis towards investigating how the Standard MPMC might be modified such that it provides one of the first three preferred output groups enumerated in Section 1.1.

1.3.1 Generic Issues

There are several generic issues relating to the use of the Standard MPMC in either risking or benchmarking that include the following:

- Training the output MPM on only 60% of patients is likely to result in a suboptimal MPM.
- Evaluating the performance of the output MPM using only 40% of patients is likely to result in an inaccurate performance point estimate.
- Logistic regression potentially lacks the versatility required to fully utilize the information contained within a large ICU Database.

Although the first two such issues are easily resolved through the use of cross-validation and the third has been addressed elsewhere (Section 2.2.2), their examination in light of the risking versus benchmarking distinction may still offer considerable insight.

1.3.2 Risking-Specific Issues

The MPM output by the Standard MPMC is not appropriate for either external or internal risking because of Step 3 (as defined in Section 1.2). For patients with ICU lengths of stay of between 4 and 24 hours the prediction time is not known until after the patient has either died or been discharged, which is clearly too late for such a prediction to be used for risking. Additionally, for patients with lengths of stay greater than 24 hours a reliable and up to date prediction can only be made after exactly 24 hours, because none of the patient profiles used to train the MPM contained information from beyond this time. However, to be useful for
1.3 Issues with the Standard MPMC

risking an MPM must be able to output reliable predictions throughout a significant portion of a patient’s ICU stay.

The second major issue with the use of such MPMs in risking relates to their predictive accuracy. The predictive accuracy of the main MPMs of the literature is potentially similar to that of nurses\(^1\). In a high-resource clinical environment such as an ICU the similar performance of mortality predictions from nurses and MPMs greatly limits the potential use of the latter as a deterioration alarm. The reason for this is that the new information with which predictions could be updated only becomes available as an ICU nurse records an event, and at such times a nurse will generally notice and alert others to abnormal events that are indicative of patient deterioration. In contrast a highly accurate MPM could offer substantial utility by providing more reliable alerts than can be generated by ICU nurses.

1.3.3 Conflation of Risking and Benchmarking Performance

This Subsection aims to disambiguate the risking and benchmarking performance of MPMs. Consider two potential measures of MPM performance; the AUC that is the primary MPM evaluator used in the Standard MPMC and the \(\chi^2\) Statistic that measures the average difference between the expected and actual number of deaths that occurred in each ICU. The issue is the extent to which each of these MPM evaluators measure each of an MPM’s risking performance and benchmarking performance. To examine this consider two simple MPMs; an ‘Age Model’ that predicts an individual patient’s risk as a function of just their age and the ‘Null Model’ that predicts all patients’ risks to be constant regardless of their clinical profile.

To understand how these two MPMs behave, two plots of age against mortality are presented in Figure 1.1 for the APACHE Database\(^2\). The left panel provides density plots as a function of age for survivors (black) and non-survivors (red). That the non-survivors tend to be older than the survivors indicates that the Age Model will predict that older patients

\(^1\)Justification for this statement is provided in Section 2.2.3
\(^2\)The APACHE Database is an ICU Database that is introduced Section 3.7
1.3 Issues with the Standard MPMC

Fig. 1.1 Plots to demonstrate the effects of age on hospital mortality for ICU patients. The left panel provides density plots of patient age for survivors (black) and non-survivors (red). The right panel plots mortality rate against mean patient age for each ICU in the APACHE Database (Section 3.7). The area of each point in the right panel is proportional to the number of patients in the associated ICU.

are more likely to die than younger ones, with such an MPM better than the Null Model if used for risking. The AUC will prefer the Age Model over the Null Model because of the appreciable separation between the age distributions for survivors and non-survivors.

The right panel in Figure 1.1 plots mortality rate against mean patient age with each point representing a different ICU. Interestingly the points in this panel exhibit negative correlation to a degree that far exceeds the differences in mortality rates that one might expect from a combination of just chance and differences in ICU quality of care. This indicates that ICUs with younger patients tend to have more severe case-mixes that ICUs with older patients. Therefore the Age Model will exacerbate the effects of case-mix and will in fact be worse than the Null Model if used for benchmarking. For similar reasons the $\chi^2$ Statistic will indicate that the Null Model is preferred over the Age Model.

That the Age Model is better than the Null Model in terms of risking but worse in terms of benchmarking suggests that different MPM evaluators are required for each of risking
and benchmarking performance evaluation. That the AUC prefers the Age Model while the $\chi^2$ Statistic prefers the Null Model suggests that the AUC may focus more on risking performance while the $\chi^2$ Statistic may focus more on benchmarking performance. If the AUC is indeed a poor measure of benchmarking performance then the Standard MPMC does not provide the full set of external benchmarking outputs as it provides a risking point error estimate rather than a benchmarking point error estimate.

1.3.4 Assuming Complete Case-mix Adjustment

This Subsection aims to demonstrate why one must account for a MPM’s benchmarking performance when interpreting its quality of care estimates. Suppose that one has an external benchmarking MPM and a set of patient profiles and mortality outcomes for an ICU for which one is interested in estimating the quality of care, and which was not one of the ICUs in the ICU Database on which the MPM was trained. Then the standard approach is to first apply the MPM to the patient profiles from the new ICU in order to generate risk estimates, which can then be compared against observed mortality rates. A ‘p-value’ is then generated as the probability that the difference between the observed and expected mortality rates would be at least as extreme as it currently is, were each patient’s actual risk equal to their predicted risk output by the MPM. A statistically significant p-value is then taken to be evidence that the ICU is providing either significantly above or below average quality of care.

However, this approach relies on the assumption that the external benchmarking MPM is able to account for case-mix in its entirety. But it seems far more likely that any real-world benchmarking MPMs will only be able to account for case-mix to a limited extent. Relaxing this assumption leads to differences in actual and expected mortality rates being attributable to a combination of differences in ICU quality of care and differences in unaccounted for ICU case-mix. Inference about ICU quality of care then requires information about the relative
1.4 Contributions

The contributions of this thesis can be split into four parts as follows:

1. The adjustment of the Standard MPMC to output appropriate benchmarking performance estimates. This includes a demonstration that alternatives to the AUC and H-L can better discriminate between MPMs for use in ICU benchmarking (Chapter 7) and a demonstration that alternative partitioning approaches are required to avoid overly optimistic benchmarking performance point estimates (Chapter 8).

2. Improving the accuracy of, and providing standard error estimates for, the performance estimates output by MPMCs. This includes the assessment of cross-validation and bootstrapping (Chapter 8) and a demonstration that the current method for constructing ICU Quality of Care p-values is flawed (Section 7.1).

3. The development and evaluation of methods for improving the risking performance of MPMs. This includes a quantification of the differences in the risking performance of MPMs expected from using each of: alternative machine learning techniques, small homogeneous versus large non-homogeneous ICU Databases, different quantities of clinical information per patient and variable versus fixed prediction times (Chapter 9).

4. The development (Section 4.9) and evaluation (Chapter 9) of the BEAST, a novel machine learning technique developed by the author that was one of the two winning entries in the Physionet/Computing in Cardiology Challenge 2012 [39], an international competition focused on ICU patient mortality prediction.
1.5 Overview of thesis

The structure of this thesis is as follows: Chapter 2 provides a literature review in which the most common MPMs and MPMCs are presented. Chapters 3-5 detail the ICU Databases and MPMC steps that are required for the investigations of the remainder of this thesis. Chapter 6 develops a set of simulations and Chapters 7 and 8 use these simulations to describe and quantitatively compare various MPM performance evaluators and partitioning approaches, respectively. Chapter 9 uses the results of Chapters 7 and 8 to investigate how different machine learning techniques, ICU Databases and prediction times affect the accuracy of MPMs for use in risking. Finally Chapter 10 details conclusions, limitations and potential further work.

Throughout this thesis, the definitions of objects denoted by Latin symbols are taken to persist across all Chapters, Sections and Subsections. In contrast the definitions of objects denoted by Greek symbols are taken to hold only within the Subsection in which they are defined, unless explicitly stated otherwise. Glossaries of persistent terms and symbols are provided in Appendices A and B.
Chapter 2

Background

The present Chapter motivates the chosen form of the Standard MPMC and examines its position within the ICU MPM literature. Section 2.1 introduces the APACHE IV Model and motivates the ways in which it was modified to form the Standard MPMC. Section 2.2 describes the MPMCs used in the development of the most widespread MPMs of the literature and how they differ from the Standard MPMC. Finally, Section 2.3 examines how the contributions of this thesis, and those of MPMs and MPMCs in general, fit into the broader field of decision support in critical care.

Throughout this Chapter the focus is placed on MPMs which utilize low frequency physiological time series, as often arise from nurse measurements, as opposed to high frequency time series recorded by autonomous bedside monitors. Such low frequency MPMs and MPMCs are favored here because they can be applied to the APACHE Database; the primary ICU Database used in this thesis.

2.1 The APACHE IV MPMC

The Acute Physiology and Chronic Health Evaluation (APACHE) Models are a set of ICU MPMs developed and maintained by Cerner Corporation. The APACHE IV Model [87] is
the fourth and most recent version of the APACHE series and is among the most accurate of the major ICU MPMs [50]. The ‘APACHE IV MPMC’ is taken to be the MPMC used to create both the APACHE IV Model and the original estimate of its generalization error, as described by Zimmerman et al. [87]. The APACHE IV MPMC was chosen as the basis for the Standard MPMC because it is accurate, well known and directly applicable to the primary ICU Database around which this thesis is based.

The structure of this Section is as follows: Subsection 2.1.1 details the steps involved in the application of the APACHE IV MPMC. Subsection 2.1.2 considers the reasons behind the chosen form of each of these steps. Finally Subsection 2.1.3 motivates how and why the APACHE IV MPMC was modified in order to form the Standard MPMC.

2.1.1 Specification

The APACHE IV MPMC proceeds through the following steps:

1. An ICU Database containing many ICU admissions (~100,000) from multiple ICUs (~100) is selected.

2. ICU admissions meeting the following exclusion criteria are removed:
   
   (a) The ICU stay was less than 4 hours in duration.

   (b) During the first 24 hours of the ICU admission at least one of the following measurements was not recorded: heart rate, respiratory rate and mean arterial pressure.

   (c) The age of the patient at ICU admission was either less than 16 years or unknown.

   (d) The patient was admitted with burns.

   (e) The ICU stay began after the patient received a transplant operation other than hepatic or renal transplants.
(f) The ICU stay began directly after the patient was transferred from a different ICU.

(g) The hospital stay corresponding to the ICU stay was greater than 365 days in duration.

(h) The ICU admission was not the first such admission of the patient within the set of ICUs and time frame covered by the ICU Database.

3. The ‘prediction time’ (the final time at which new information can be used to assess a patient’s risk) is set to be between 4 and 24 hours, depending on a patient’s ICU length of stay. For patients with an ICU length of stay greater than 24 hours, prediction times are set to be 24 hours after ICU admission. For patients with an ICU length of stay of between \(4\theta\) and \(4\theta + 4\) hours, for \(\theta \in \{1, 2, 3, 4, 5\}\), prediction times are set to be \(4\theta\) hours after ICU admission.

4. Patients are randomly partitioned into a ‘training set’ containing 60% of patients and a ‘test set’ containing 40% of patients, without regard to how patients are distributed among ICUs.

5. A small number (~30) of clinically relevant features are heuristically selected and then extracted. For various physiological features only the ‘worst’ (highest or lowest depending on the feature) values are retained. The full details of this step are provided in Section 3.4.

6. Physiological features are combined to form an Acute Physiology Score (APS) as described in Section 3.4.1.

7. The APACHE IV Model is trained by applying unregularized logistic regression with spline terms, as described in Section 4.5, to the patients in the training set.
8. The APACHE IV Model is used to make predictions on the test set and the Area Under the Curve (AUC) and Hosmer-Lemshow Statistic (H-L) are evaluated on these predictions in order to generate performance point estimates.

9. The APACHE IV Model and the performance point estimates are output.

2.1.2 Motivation

The choice of ICU Database in Step 1 matches the intended use of the APACHE IV Model as an external benchmarking MPM. Had only a small number of ICUs been included then their average quality of care might be substantially above or below that of the population of ICUs in which one is interested. Comparing the expected and actual mortality rates for new ICUs of interest could then give rise to biased quality of care inferences.

The exclusions in Step 2 are a combination of the removal of admissions with insufficient information to make a reliable risk prediction (a-b), the removal of patients whose physiological response is expected to be fundamentally different to the majority of other patients (c-f) and removals that are necessary for model development (g-h). The second of these three exclusion types may only be necessary because of the inflexibility of logistic regression models.

The variable prediction times of Step 3 are viable because the APACHE IV Model is intended for use in benchmarking and predictions can therefore be made retrospectively. Effectively each retained ICU admission is assigned to one of six groups depending on its duration: 4-8 hours, 8-12 hours, 12-16 hours, 16-20 hours, 20-24 hours or greater than 24 hours. Each group then has its features evaluated over a different time frame, after which all of the groups are recombined. This separation into groups both provides more accurate risk predictions for longer ICU stays and allows the retention of short stays, while the recombining of groups ensures that only a single APACHE IV Model is needed to make predictions on admissions corresponding to any of the six ICU stay groups.
2.1 The APACHE IV MPMC

The partitioning approach of Step 4 creates only one training set and one test set. Such single partitions can be extremely useful for developing heuristic models as they allow researchers to set aside the test set and then iteratively update and refine their models using only the training set. Once they are satisfied with the MPM that they have produced using only the training set they can then apply it to the patients in the test set in order to generate an unbiased performance point estimate. In contrast, using multiple partitions, such as with cross-validation, requires the researcher to fully specify the form of the MPMs that he or she intends to train before examining any patients in detail.

The use of ‘worst’ physiological values as features in Step 5 is to minimize the extent to which interventions mask poor health. Generally if a patient presents abnormal vital signs, such as high blood pressure, then they will be prescribed some form of corrective treatment, such as anti-hypertensives, in order to stabilize their condition. After such treatments the patient’s vital signs may return to normal, but the patient will often retain some underlying condition that is predictive of mortality. The use of worst values, as opposed to, say, final values, ensures that such conditions are not masked by treatment.

The use of the APS in Step 6 provides a range of benefits including placing a limit on how much a single outlier can influence a given risk prediction and ensuring both ease of evaluation and interpretation. A further benefit is that the APS can account for non-linear influences of a patient’s physiology on their risk, such that the assumptions of linear models like logistic regression become less problematic.

The use of logistic regression in Step 7 ensures that the APACHE IV Model is both easy to interpret and of a form that has already been accepted throughout the ICU community, with all of the preceding widespread ICU MPMs also having been trained using logistic regression. The lack of flexibility associated with logistic regression is mitigated through the use of spline terms in Step 7, the adoption of the APS in Step 6 and the additional exclusion criteria (c-f) of Step 2. Meanwhile the lack of regularization does not pose problems because
2.1 The APACHE IV MPMC

of the heuristic selection of features in Step 5, wherein expert knowledge is effectively used to replace regularization, in addition to the large number of patients included in the ICU Database on which the APACHE IV Model is trained.

Presenting generalization error estimates in Step 8 as AUC and H-L values ensures ease of comparison with previous MPMs, the majority of which were evaluated in the same way. By contrast, the use of the creator’s own performance measure to demonstrate that their MPM is superior to that of existing MPMs would be naturally met with considerable skepticism.

2.1.3 Adaptation

The APACHE IV Model is a well-defined MPM that can be applied to any ICU Database containing sufficient information in an unambiguous manner. However, the APACHE IV MPMC is much less well defined because it was intended as a description of how the APACHE IV Model was trained rather than a general set of instructions for how similar MPMs could be trained using other ICU Databases. This then motivates the introduction of the Standard MPMC as an alternative to the APACHE IV MPMC, one that is more readily applicable to new ICU Databases but that is otherwise as similar to the APACHE IV MPMC as possible. This Subsection details and motivates the alterations applied to the APACHE IV MPMC in order to create the Standard MPMC.

Suppose that one has a new ICU Database containing a patient population that differs considerably from that of the patients used to train the APACHE IV Model. Such patients might be from 20 years after the APACHE IV Model was trained or might be from ICUs in a third world country, such that the demographics and treatments in the ICU Database to which the MPMC is to be applied differ considerably from those in the 2002-2003 cohort of US patients used to train the APACHE IV Model. Then even if the physiological features contained within such an ICU Database align with those that contribute to the APS, issues
will still arise when applying the APACHE IV MPMC because of the inclusion of the APS and spline terms as well as the lack of regularization.

The APS is effectively an MPM trained on an ICU Database that predates the ICU Database used to train the APACHE IV Model. It will therefore become less able to account for patient physiology as it is applied to patients who are less similar to the patients in the ICU Database on which the APS was developed. The effect, known as ‘model drift’, would then hinder the application of the APACHE IV MPMC to new ICU Databases. One way to overcome APS model drift is to retrain the APS on the ICU Database to which the APACHE IV MPMC is applied. Methods for achieving this are provided in Johnson et al. [41], however this approach adds a considerable degree of complexity to an MPMC. Instead the intermediate step of combining the worst physiological values of each type into the APS and then using the APS as one of the features with which the logistic regression model is trained can be replaced by feeding all of the worst physiological values directly into the logistic regression model. Ideally the ‘worst’ physiological values, which are defined relative to the APS, would be replaced with both maximum and minimum values. However, such maximum and minimum values have not been included in the Standard MPMC because of their absence from the primary ICU Database on which this thesis is based.

The removal of the APS has several undesirable effects that must be addressed. The first is that outliers become far more problematic with the removal of the effective limit on the extent to which a single outlier can influence any single risk estimate. A simple form of automated preprocessing is therefore included in the Standard MPMC in order to resolve this issue, as described in Section 3.6. The second effect is that without the APS any non-linear influences of physiological measurements on a patient’s risk can no longer be accounted for using an unmodified logistic regression model. This is clearly an issue because most physiological values have some natural range with extremely high or low values indicative of poor health. This issue is resolved by allowing the logistic regression model implemented in
the Standard MPMC to take as inputs both linear and square terms as opposed to just linear terms, as described in Section 4.4. The addition of square terms also diminishes the need for spline terms which are therefore left out of the Standard MPMC. The exclusion of spline terms was also motivated by the heuristic nature of the selection of the number and location of spline knots in the APACHE IV MPMC.

The ICU Database on which the APACHE IV Model was trained contained a large number of patients and only a small number of features. The use of regularization to reduce overfitting when training the APACHE IV Model was therefore unnecessary. However, replacing the APS with a variety of different types of physiological values increases the number of features, while there is no guarantee that one would intend to only apply the Standard MPMC to ICU Databases containing very large numbers of patients. The Standard MPMC is therefore chosen to implement regularized logistic regression in place of unregularized logistic regression, as described in Section 4.4.

2.2 Literature Review

This Section examines the MPMCs of the MPM literature, with the primary focus on the documentation of potential alternatives to the steps of the Standard MPMC. These will then be combined with further alternatives introduced by the author and quantitatively compared with each other and the steps of the Standard MPMC throughout the remainder of this thesis. For convenience this Section splits the MPM literature into the following four parts and considers each in turn:

1. Construction: Investigations that apply an MPMC to an ICU Database in order to output an MPM and an estimate of its generalization error.
2.2 Literature Review

2.2.1 MPM Construction

A construction investigation constitutes an article or paper that applies an MPMC to an ICU Database, in order to output an MPM and an estimate of its generalization error. In this Subsection a total of ten key construction papers are presented in terms of how the MPMC that they apply differs from the Standard MPMC. A summary of the MPMs output by these construction papers is presented in Table 2.1.

The APACHE Models

The Acute Physiology and Chronic Health Evaluation (APACHE) Models [48, 45, 47, 87] are a set of MPMs trained on United States ICUs. The fourth and most recent version of these models (APACHE IV) is perhaps the most accurate of the widely-used ICU MPMs [50]. The MPMC used to train the APACHE IV Model (detailed in Section 2.1.1) and the MPMCs used to train the earlier versions of the APACHE Model are very similar, except that the older MPMCs had fewer features and excluded spline terms. The original APACHE Model was the first, and the APACHE II Model was perhaps the most widely used, of the major ICU MPMs [75]. The development of the APACHE III Model included the creation of the Acute Physiology Score, which forms a part of the APACHE IV Model.
Table 2.1 Table summarizing the most commonly used ICU MPMs. The prediction time is measured relative to ICU admission. The two performance measures are the Area Under the Curve (AUC) and the Hosmer-Lemeshow Statistic (H-L), as described in Section 7.2.1. A larger AUC indicates superior discrimination while a smaller H-L indicates superior calibration. The performance values are the internal validation values reported in each model’s original publication.

The SAPS and tMPMs

The Simplified Acute Physiology Score (SAPS) Models [51, 56, 57] are a series of MPMs that began as a simplification of the original APACHE Model and that increased in complexity in subsequent versions. SAPS II was trained on European and North American ICUs while SAPS III was trained on ICUs from various continents. These two versions of SAPS differ in terms of their prediction times, with SAPS II utilizing information recorded during the first 24 hours of the patient’s ICU stay, while SAPS III does not use information recorded beyond one hour after ICU admission.

The Mortality Prediction Models [52, 36] (tMPMs) are a series of MPMs trained on European and North American ICUs. The MPMCs used to train the three versions of tMPM considered differ in terms of their prediction times, as presented in Table 2.1, but are otherwise very similar.

The main difference between the Standard MPMC and the MPMCs used to train the various versions of SAPS and tMPM relates to their prediction times. The earlier prediction
times of SAPS III, tMPM\textsubscript{0} II and tMPM\textsubscript{0} III were introduced to reduce the Boyd and Grounds effect; in which an ICU that provides exceptional quality of care during the first 24 hours will tend to suppress the signs of ill health within its patients more than average, such that an MPM with a large prediction time assesses its case-mix to be less severe than it actually is and therefore resulting in an unduly poor quality of care evaluation. However, reducing the prediction time also reduces the information available to an MPM with a corresponding reduction in accuracy [9].

**The ICNARC Model**

The Intensive Care National Audit and Research Council (ICNARC) Model [33] is an MPM trained exclusively on ICU patients from within the United Kingdom. Such patients tend to be more severely ill than those admitted to US ICUs [6] and as such the ICNARC model should not be applied to US ICU patients without prior modification. However, the MPMC used to generate the ICNARC Model warrants consideration because of the innovative partitioning approach that it implements.

In the Standard MPMC patients are randomly partitioned into a single training set and test set. In contrast the ICNARC MPMC performs cross-validation over ICUs, wherein ICUs are split into multiple groups. For each such group an MPM is trained on all of the patients belonging to ICUs outside of that group, with the MPM then used to make predictions for the patients inside the group. The use of cross-validation instead of holdout ensures that every patient is represented in both training and testing sets in order to reduce uncertainty, while the splitting over ICUs as opposed to patients potentially helps ensure that the MPM does not produce overly optimistic benchmarking performance estimates by unduly accounting for ICU quality of care.
The ANZROD Model

The Australian and New Zealand Risk of Death (ANZROD) Model [62] is an MPM trained exclusively on Australian and New Zealand ICU patients and is perhaps the newest major MPM to date. The MPMC used to train the ANZROD Model, the ‘ANZROD MPMC’, is a good example of a modern MPMC, and understanding the motivation behind its choices may provide considerable insight.

ICUs in Australia and New Zealand receive guidance from the Australian and New Zealand Intensive Care Society (ANZICS). To enable ANZICS to provide ICU benchmarking, ICUs are required to provide ANZICS with the patient information required to evaluate the APACHE III model [24]. This information has then been collated to form an ICU Database known as the ANZICS Adult Patient Database [74]. The ANZICS Adult Patient Database is substantial, containing information from hundreds of thousands of ICU stays from well over 100 Australian and New Zealand ICUs, and represents an excellent resource with which to train MPMs. In 2012 the effect of model drift, wherein the predictions output by MPMs become less accurate over time due to changes in ICU treatments and ICU patient case-mix, was demonstrated for the APACHE III model applied to Australian and New Zealand ICU patients [63]. This then motivated the training of an updated MPM on the ANZICS Adult Patient Database, with ANZROD as the result [62].

The ANZROD MPMC differs from the Standard MPMC in two major ways: The first is that instead of using the APACHE IV features, the APACHE III features are used in addition to the length of hospital stay prior to ICU admission and treatment limitations [29]. The second is that different logistic regression parameters are fitted for patients belonging to each of several different major disease groups. Both the Standard MPMC and the ANZROD MPMC apply logistic regression and split patients into a single training set and a single testing set without regard to ICU memberships. They also both use the AUC and the H-L
Statistic as their primary methods for evaluating the performance of the MPMs that they output, although the ANZROD MPMC also provides AUC confidence intervals.

The ways in which the ANZROD MPMC differs from the Standard MPMC indicates various issues that the authors identified with the application of the APACHE III MPMC to the ANZICS Adult Patient Database. The first relates to the extent to which the choice of features was appropriate, with such concerns naturally depending on the ICU Database to which an MPMC is to be applied. Often there may be features for which information is not present, which naturally motivated the use of the APACHE III features in the ANZROD MPMC rather than switching to the more accurate set of APACHE IV features. Alternatively, there may be additional information beyond that which is exploited by the features in the Standard MPMC, which motivated the inclusion of treatment limitations in the ANZROD MPMC. The second difference, in which different sets of logistic regression parameters were trained for different major disease groups, alludes to the idea that logistic regression models are too inflexible to cope with the diverse range of conditions from which ICU patients may suffer.

### 2.2.2 MPM Comparison

MPM comparisons involve the application of an MPMC to an ICU Database to generate multiple MPMs through different training methods, in order to determine which training method is superior. At their core these investigations can be thought of as MPMCs that apply multiple instances of the Standard MPMC, each with a different machine learning technique replacing logistic regression. The best technique is usually stated to be the one with the greatest discrimination (AUC), out of all the techniques that are well calibrated (H-L Statistic with an associated p-value of greater than 0.05). This approach is therefore taken as the baseline for MPMCs that compare MPM training methods.
Comparison papers often replace the division of patients into a single training set and a single test set with cross-validation. It is also common for p-values to be output as an indication of the statistical significance associated with differences in AUCs. Several papers have compared logistic regression against more advanced machine learning techniques. Wong and Young [85] found no significant difference between the performance of neural networks and the APACHE II Model on an ICU Database of 8,796 patients. Verplancke et al. [79] found no significant difference between the performance of support vector machines and logistic regression on an ICU Database of 352 patients with haematological malignancies. Pirracchio et al. [65] found that an approach that combines a range of machine learning techniques to significantly outperform SAPS II on an ICU Database of 24,508 patients. Clermont et al. [17] found no significant difference between the performance of neural networks and logistic regression on an ICU Database of 1,647 patients, while Jaimes et al. [38] found neural networks to significantly outperform logistic regression on an ICU Database of 533 sepsis patients.

Comparisons may extend to encompass different sets of features in place of different machine learning techniques. One such paper by Bukan et al. [23] found that an MPM based on pre-admission quality of life offers similar predictive accuracy to the APACHE II model, suggesting a great deal of potential within ICU benchmarking. Comparison may also include a combination of new features and new machine learning techniques, such as that by Ghassemi et al. [28], indicating the utility of having MPMs utilize free text hospital notes.

One comparison paper of particular interest is that by Ramon et al. [68], who used an ICU Database containing 1,548 patients to compare the mortality predictions output by different machine learning techniques with the predictions of nurses and doctors. They found that the predictions of doctors were superior to those of nurses, and that the predictions output using Naive Bayes and Random Forests were comparable to those of nurses and doctors, respectively.
2.2 Literature Review

2.2.3 MPM Validation

Validation studies assess the performance of one or more MPMs on external ICU Databases. APACHE II, APACHE III, tMPM₀ II, tMPM₂₄ II, SAPS II and SAPS III have each undergone extensive validation [75], while the APACHE IV model has undergone moderate validation [13, 50, 44]. The ICNARC and ANZROD models are difficult to externally validate because they are both specifically designed for use in ICUs within a given set of countries and were trained using the vast majority of such ICUs. Meanwhile tMPM₀ III has generally received far less attention than tMPM₀ II. The results of such validation studies tend to favor more recently published models [32], especially the APACHE models. As differences between the populations on which models are trained and validated increase, model discrimination (AUC) tends to undergo only a modest degree of deterioration while model calibration (H-L) tends to degrade rapidly [58].

Some validation studies also include measures of the accuracy of mortality predictions made by physicians, with the aim of comparing the predictive performance of physicians and MPMs. However, setting up such a study so that physicians and MPMs are compared fairly has proven difficult. The most recent meta-study found that physicians greatly outperform MPMs [73], although it is acknowledged by the study’s authors that this conclusion is weakened by the small number and ‘moderate methodological quality’ of available studies. There is also the issue that the meta-study, through necessity, reports the average performance over a variety of different MPMs rather than that for a single accurate MPM, such as the APACHE IV Model. Further, there is evidence that the accuracy of mortality predictions made by doctors greatly exceeds that of mortality predictions made by nurses [46]. Given the contradictions between the findings of this meta-study and the work of Ramon et al. [68], there does not appear to be consensus on the relative accuracy of the APACHE IV Model and nurses.
2.2.4 MPM Review

Several excellent review articles exist that cover both the main construction papers and the validation of the MPMs that they output [75, 43, 82]. Though the main contents of these reviews have been covered in this Chapter it is noted that the summary of validation papers provided in Table 2 of Strand and Flaatten [75] represents a sizable addition.

Several articles focus on the potential issues associated with the use [83, 10] and evaluation [71] of MPMs in risking. The most significant of these reported issues were described and addressed in Section 1.3, and go on to motivate some of the alternative MPMC steps considered later in this thesis.

2.3 Machine Learning in Critical Care

The previous Sections of this Chapter examined a variety of important MPMs and MPMCs. In the present Section the focus is broadened to consider how these MPMs and MPMCs, as well as the contributions of this thesis as a whole, fit into the broader field of decision support in critical care. ‘Machine Learning and Decision Support in Critical Care’, a comprehensive review of decision support in critical care by Johnson et al. [40], examines and categorizes the major challenges preventing ICU decision support systems from realizing their potential. The present Section offers a precis of this work, outlining the three broad categories of challenge, and identifying the specific challenges within these categories that the present thesis aims to help address.

Compartmentalization

Compartmentalization is taken to encompass the various barriers that must be overcome before ICU patient information can be accessed in a timely manner. One such barrier relates to the various medical data protection acts [14] that have been put in place in order to protect
patient privacy. The utility of such safeguards is clear, as the public disclosure of patients’ medical history in an identifiable manner would cause a great deal of distress. However, such protocols inevitably make accessing patient data for research purposes more difficult [59]. The other two barriers that must be addressed within compartmentalization are what Johnson et al. refer to as ‘integration’ and ‘harmony’. Integration must be performed on the disparate sources of information output by different medical devices [53] while harmony requires data definitions within different institutions to be made homogeneous [21].

Addressing each of these three aspects of compartmentalization undoubtedly represented an enormous amount of work for those who compiled both the APACHE IV and MIMIC II Databases around which this thesis is built. Both databases underwent several de-identification steps and required both training and the signing of a data use agreement in order to gain data access. However, the author neither contributed to, nor had any need to reproduce any of the solutions to compartmentalization that were applied in order to form these databases, and the processes surrounding compartmentalization fall well outside of the author’s expertise. Compartmentalization has therefore been placed outside of the scope of this thesis, in spite of its central role within machine learning in critical care.

**Corruption**

Corruption is taken to encompass the issues relating to data quality that remain after the data has been unified. One issue is that of erroneous data arising from a variety of sources including human error, such as from an incorrectly transcribed blood pressure reading or the miscounting of breaths when measuring respiratory rate, and instrumentation problems, such as an incorrect heart rate reading due to artifacts caused by patient movement or by an equipment fault. The standard approach is then to remove extreme values either through the application of domain knowledge [87] or through statistical means [5]. Another issue relates to missing data, where the information required to evaluate a feature of interest is sometimes
available for only a subset of patients. Missing data is a common issue in data science and has received considerable attention [54]. The issue is particularly prevalent in clinical settings as the primary motivation behind data collection is usually to support individual patient treatment decisions rather than for research purposes, such that some measurements are made only if a physician expects the outcome to inform their treatment choices for the associated patient [40]. In spite of this, it seems that missing data is often poorly addressed in clinical care research [80].

The degree to which an MPM must be robust to data corruption depends to some extent on its intended use. To function appropriately, a risking MPM must be embedded within an ICU in such a way that it can make continuous up-to-date risk estimates by accessing a patient’s electronic medical records as these records are updated. Such MPMs will inevitably encounter the two forms of corruption considered above and will need to cope with them in a robust manner, such that reliable predictions can be made without the need for further human input. In contrast, benchmarking MPMs tend to be applied retrospectively such that the data that they use can undergo some form of manual quality control if necessary. However, even for benchmarking it would be preferred if MPMs could resolve data corruption issues without additional human input, because such input could give rise to further sources of bias.

Data corruption is an important issue that must be addressed when considering any MPM and such methods are provided in this thesis. However, the APACHE Database underwent a significant degree of preprocessing before it became available to the author (Chapter 3.7). As such it does not represent a good resource with which to investigate the extent to which different preprocessing approaches are more or less appropriate than one another for resolving ICU Database corruption. The approach is then to take preprocessing as a necessary part of this thesis, but not one of its focuses.
Complexity

Complexity is taken to encompass the difficulties surrounding the creation of useful [66] and reliable outputs from preprocessed ICU data. Complexity relates to the idea that the trajectory of a human’s physiology is often difficult to predict and that this is further compounded by the vast range of conditions from which ICU patients suffer. In their review, Johnson et al. [40] present various machine learning techniques that have been tailored to cope with complexity in ICU data, and several of these techniques were considered earlier in this Chapter. They also provide examples of clinical validation, with one such study of an early warning system by Clifton et al. [19] being of particular interest. However, the general focus of this thesis within the topic of complexity deviates from that of Johnson et al. by instead examining MPMCs as a framework with which to efficiently evaluate different combinations of machine learning techniques, feature sets and preprocessing approaches, with the intention that the most appropriate MPMs for each of benchmarking and risk scoring for a given scenario might more readily be selected.
Chapter 3

Data Preparation

This Chapter focuses on ICU Databases and their conversion into a form appropriate for machine learning. This Chapter has three aims, the first being to introduce the ICU Databases used in this thesis (Sections 3.1, 3.7 and 3.8). The second is to detail how an appropriate set of patient profiles and responses can be extracted from a generic ICU Database (Section 3.2, 3.3 and 3.5). The third is to provide tools for converting such a set of patient profiles into a ‘design matrix’ that is ready for the application of machine learning (Sections 3.4 and 3.6). Details of the historical [22] and state of the art [27] methods used to extract ICU Databases fall outside the scope of the present work.

3.1 Events and ICU Databases

Let an ‘ICU stay’ refer to an uninterrupted interval during which a given patient remains in a single ICU. Then an ICU Database will usually contain information on all, or some explicit subset of, ICU stays occurring within some fixed set of ICUs over some fixed period of time. Generally, this information will be recorded as a large number of ‘events’, each composed of five values, \( \{e^1, e^2, e^3, e^4, e^5\} \), for which:
• $e^1$ is the type of event, such as a heart rate measurement or age assessment. In this thesis the values for $e^1$ of 0, 1, 2, 3, 4 and 5 are reserved for patient death, hospital admission, ICU admission, ICU discharge, hospital discharge, and primary ICD9 Code\(^1\), respectively.

• $e^2$ is the time-stamp for when the event occurred, such as when a measurement was taken, when a medication was received or when a patient died.

• $e^3$ is the time-stamp for when the event was recorded; this will generally take the same value as $e^2$ except where measurements have substantial processing times, such as with laboratory tests.

• $e^4$ is the value of the record, such as the heart rate in beats per minute, the gender as male/female, the dose in micro-grams or the index of the ICU to which the patient was admitted.

• $e^5$ is the index of the patient to whom the event relates.

The events can be split into the eight following classes:

1. Instantaneous measurements of a patient’s state, such as a heart rate measurement or blood pressure measurement.

2. Obvious changes in a patient’s state, such as death\(^2\).

3. Cumulative measurements, such as the volume of urine output since the last time the same type of measurement was taken.

4. Demographic records, such as a patient’s date of birth or gender\(^3\).

\(^1\)For a patient that dies in hospital the primary ICD9 Code usually reports the disease that is thought to be the primary cause of the patient’s death.

\(^2\)These are effectively instantaneous measurements that occur continuously.

\(^3\)These are effectively instantaneous measurements for which the result is assumed to remain constant throughout a patient’s hospital stay.
5. Instantaneous treatments, such as the administration of a single dose of a particular drug.

6. Transitions between continuous treatments, such as to or from mechanical ventilation.

7. Transfers between care environments, such as from an ICU to the hospital floor.

8. A diagnosis, such as epilepsy or appendicitis.

An ICU stay begins when a patient is either admitted to an ICU from a non-ICU hospital department\(^4\) (herein referred to as the ‘hospital floor’) or transferred from a different ICU within the same hospital. An ICU stay ends when a patient is either discharged from the ICU to the hospital floor\(^5\), transferred to another ICU within the same hospital, or dies.

Each ICU stay is a subinterval of a ‘hospital stay’; an uninterrupted interval during which a given patient remains in a single hospital. A hospital stay begins when a patient is either admitted to a hospital from outside of hospital care or transferred in from another hospital. A hospital stay ends when a patient is either discharged from hospital care, transferred to another hospital, or dies.

A set of example events for a single fictitious hospital stay is provided in Table 3.1. In this example the patient is admitted to an ICU 5 hours after hospital admission. The patient then remains in this ICU for 40 hours, during which time they have their gender, white blood cell count and several blood pressure measurements recorded, and are administered anti-hypertensives. The ICU stay ends when the patient is discharged to the hospital floor for 11 hours, after which they are transferred to another ICU. The patient is then discharged to the hospital floor and subsequently discharged from hospital, with their primary ICD9 Code recorded as epilepsy; they die 9 days later.

\(^4\)For simplicity transfers from both step-down units and direct ICU admissions will be treated as passing though the hospital floor.

\(^5\)Again for simplicity transfers to both step-down units and directly home from the ICU will be treated as passing though the hospital floor.
Table 3.1 Example events for a single fictitious hospital stay. The first through fifth columns provide the type of event, the time-stamp for when the event occurred, the time-stamp for when the event was recorded, the value associated with the event and the patient to which the event corresponds, respectively.

<table>
<thead>
<tr>
<th>$e^1$</th>
<th>$e^2$</th>
<th>$e^3$</th>
<th>$e^4$</th>
<th>$e^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Hospital admission)</td>
<td>95</td>
<td>95</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2 (ICU admission)</td>
<td>100</td>
<td>100</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>6 (Gender)</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>100</td>
<td>100</td>
<td>156</td>
<td>1</td>
</tr>
<tr>
<td>8 (Hydrochlorothiazide)</td>
<td>100.2</td>
<td>100.2</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>104.2</td>
<td>104.2</td>
<td>133</td>
<td>1</td>
</tr>
<tr>
<td>9 (Urine output)</td>
<td>110.0</td>
<td>110.0</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>112.6</td>
<td>112.6</td>
<td>120</td>
<td>1</td>
</tr>
<tr>
<td>10 (White blood cells)</td>
<td>120.2</td>
<td>126.4</td>
<td>5160</td>
<td>1</td>
</tr>
<tr>
<td>8 (Urine output)</td>
<td>134.0</td>
<td>134.0</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>136.7</td>
<td>136.7</td>
<td>118</td>
<td>1</td>
</tr>
<tr>
<td>3 (ICU discharge)</td>
<td>140</td>
<td>140</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>2 (ICU admission)</td>
<td>150</td>
<td>150</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>3 (ICU discharge)</td>
<td>152</td>
<td>152</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>4 (Hospital discharge)</td>
<td>160</td>
<td>160</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>5 (Primary ICD9 code)</td>
<td>160</td>
<td>160</td>
<td>345</td>
<td>1</td>
</tr>
<tr>
<td>0 (Death)</td>
<td>385</td>
<td>385</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>
3.2 Patient Profiles

A patient profile is a list of events that can serve as the input to an MPM. To convert a list of events into a patient profile it is necessary to obscure various event-related information. This includes all events that became available after the time at which a prediction is to be made (Section 3.2.1) and events that could excessively aid inference about the ICUs to which patients are assigned (Sections 3.2.2-3.2.3).

3.2.1 Prediction Time Exclusions

For risking, the ‘prediction time’ for a given ICU stay specifies how long after ICU admission the associated patient’s risk is to be estimated. For benchmarking, the prediction time specifies which events are used for making predictions, with all events that occurred after prediction time removed. From an implementation standpoint, the two definitions of the prediction time are equivalent.

A prediction time of zero indicates that only information recorded at or prior to ICU admission may be used to make predictions. Such an early prediction time can be beneficial for benchmarking, because predictions can be made for all ICU stays and there is no feedback bias caused by better ICU treatment indicating that a patient is less unwell than they actually were at ICU admission, and vice versa. However, larger prediction times allow for more information about a patient to be gathered, and so more accurate predictions can be made. This trade-off has led to the implementation of a range of prediction times within the ICU benchmarking literature between 0 and 24 hours, as was presented in Table 2.1.

The ‘length of stay’ associated with a given ICU stay is taken to be the time between ICU admission and ICU discharge. Let the length of stay and prediction time for the $i$’th ICU stay be denoted $T_i$ and $t_i$, respectively. Further let a prediction time inducer be a function that maps a length of stay onto a prediction time. Then in this thesis a total of three prediction time inducers are defined as follows:
3.2 Patient Profiles

1. The Fixed Prediction Time Inducer; \( t_i = 24 \) hours for all \( i \).

2. The Deterministic Variable Prediction Time Inducer; for \( \theta \in \{1, 2, 3, 4, 5\} \), \( t_i = 4\theta \) hours if \( 4\theta < T_i \leq 4\theta + 4 \), and \( t_i = 24 \) hours otherwise.

3. The Stochastic Prediction Time Inducer; each \( t_i \) is drawn from a uniform distribution between 24 hours and \( T_i \).

The Deterministic Variable Prediction Time Inducer was used to train the APACHE IV Model, with shorter prediction times used for ICU stays with a length of stay less than 24 hours. The Stochastic Prediction Time Inducer is introduced to help with risking, as this usually requires an MPM that can output continuous predictions, as is clearly the case for patient deterioration alarms. When training such an MPM it is important to use a range of prediction times that resemble those to which the MPM is likely to be applied, and the Stochastic Prediction Time Inducer represents a straightforward example of this.

For risking, all events that were unavailable at the prediction time \( (e^3 > t_i) \) must be removed from a patient’s profile. For benchmarking, predictions are not required until after prediction time and it is therefore possible to remove only events for which \( e^2 > t_i \). However, the values of \( e^2 \) and \( e^3 \) tend to be extremely similar and for convenience the equality \( e^3 > t_i \) is used for event exclusion for both risking and benchmarking throughout the remainder of this thesis.

3.2.2 Transfer Event Obfuscation

For benchmarking, it is necessary to stop an MPM accessing events that would provide too much information about the ICU to which a given patient is assigned. Otherwise the MPM may make predictions that account for ICU quality of care, such that the quality of care cannot be inferred from differences between the actual and estimated ICU mortality rates. For such events it is sufficient to mask the recorded values \( (e^4) \) associated with transfer.
events \(e^1 \in \{1, 2, 3, 4\}\), as these values identify the transfer destination. For convenience, this masking is performed for both risking and benchmarking, with the assumption that the effect of differences in ICU quality on patient mortality rates will be small compared to those caused by differences in patient physiology.

### 3.2.3 Treatment Event Retention

Treatment events provide information about the care that a patient receives. For risking, the merit of including such events is clear as one is interested in the extent to which a patient is in danger, even if the cause is related to the patient’s treatment.

For benchmarking, the choice to include treatment events is less obvious as one is only interested in how ill a patient is at ICU admission. Including treatment events for benchmarking MPMs may result in the output predictions accounting for poor treatment choices, such that ICUs are not penalized for poor treatment choices made early in a patient’s ICU stay. However, excluding treatment events can lead to biased and inaccurate predictions as MPMs are left to infer the meaning behind events without the appropriate context. Consider the admission detailed in Table 3.1. Here the patient is admitted with high blood pressure, prescribed anti-hypertensives, and has their blood pressure reduced as a result. Considering only blood pressure at the end of day 1 would indicate nothing wrong with the patient, whereas considering blood pressure and its treatment would provide the correct diagnosis of a somewhat unwell patient. If the patient had instead been admitted to an ICU that failed to respond to their elevated blood pressure, then at the end of day 1 the patient would likely have retained a very high blood pressure. Examining a patient profile without treatment events might then lead to the assumption that the patient has had anti-hypertensives administered and shown no response to this treatment. The poor treatment decision would then suggest

---

6It is noted that a viable alternative is to allow \(e^d\) for \(e^1 \in \{2, 3\}\) to contain the type of ICU rather than the ICU’s unique identifier. This would then allow MPMs to include a single offset per ICU type in order to help account for the large differences in case-mix that these distinctions might introduce.
that the ICU had received a patient who was far more unwell than they actually were, such that the ICU might be seen as performing fine in spite of its actions causing a potentially elevated mortality rate\(^7\). On the other hand, including treatment events would make the predicted mortality similar for both good and poor treatment, as it would become obvious that hypertension was initially missed and was likely to be noticed and treated shortly after, assuming that the MPM did not infer that the patient without treatment was more likely to die because they were in a poorly performing ICU.

Throughout the remainder of this thesis, treatment events are included for both benchmarking and risking MPMs. This is in line with the Standard MPMC, which includes mechanical ventilation and intubation as features. However, it is noted that it is still an open question as to whether this is the best approach for MPMs that are to be used in benchmarking. Notably the distinction between treatment events and non-treatment events can be exploited to make interesting inferences. An example is where a treatment for a new patient is flagged as a potential error if the treatment provided differs from that previously administered to patients with a similar physiology\(^{35}\). In certain situations, a robust MPM incorporating treatment events might similarly flag a treatment choice as bad if it significantly raises a patient’s predicted mortality risk.

An example patient profile is provided in Table 3.2, obtained through the application of the Fixed Prediction Time Inducer and Transfer Event Obfuscation to the ICU Database events for the first ICU admission in Table 3.1. Such a patient profile can then be used as the input to an MPM, or a set of such patient profiles and their corresponding responses can be used as the input to an MPMC in order to train an MPM.

\(^7\)Notably the use of the ‘worst’ recorded values as in the APACHE IV Model greatly mitigates this issue as the worst physiological values will refer to those recorded before the treatment-induced improvements.
### Table 3.2 List of events forming the patient profile, generated by the application of the Fixed Prediction Time Inducer and Transfer Event Obfuscation to the ICU Database events for the first ICU admission in Table 3.1. The first through fifth columns provide the type of event, the time-stamp for when the event occurred, the time-stamp for when the event was recorded, the value associated with the event and the patient to which the event corresponds, respectively.

<table>
<thead>
<tr>
<th>( e^1 )</th>
<th>( e^2 )</th>
<th>( e^3 )</th>
<th>( e^4 )</th>
<th>( e^5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Hospital admission)</td>
<td>-5</td>
<td>-5</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>2 (ICU admission)</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>6 (Gender)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>0</td>
<td>0</td>
<td>156</td>
<td>1</td>
</tr>
<tr>
<td>8 (Hydrochlorothiazide)</td>
<td>0.2</td>
<td>0.2</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>4.2</td>
<td>4.2</td>
<td>133</td>
<td>1</td>
</tr>
<tr>
<td>9 (Urine output)</td>
<td>10.0</td>
<td>10.0</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>7 (Systolic blood pressure)</td>
<td>12.6</td>
<td>12.6</td>
<td>120</td>
<td>1</td>
</tr>
</tbody>
</table>

#### 3.3 Response Selection

For risking, one is interested in using a patient profile to assess a patient’s overall state of health, such that alerts can be generated on the patient for which that measure is deteriorating. The approach used throughout this thesis is to estimate the probability of patient death before hospital discharge, and then to use this estimated probability as a surrogate for the patient’s overall health. Here death before hospital discharge is an example of a ‘response’: something that is to be predicted using the patient profiles. For ICU benchmarking the use of the response is even more direct, with ICU performance derived from differences between the expected and actual number of patients who exhibited the response.

The aim of this Section is to justify the use of death before hospital discharge as the primary response for ICU MPMCs. The potential pros and cons associated with this and other potential responses are provided, although notably the alternatives to death before hospital discharge are not implemented at any stage during this thesis.
3.3 Response Selection

Death before hospital discharge

The response used in the Standard MPMC and throughout the remainder of this thesis is death before hospital discharge. Two clear benefits of this response are that both death and hospital discharge are extremely well defined, and that death is a clear indication of poor health. The use of this response also makes comparisons with existing MPMs more straightforward as these too tend to use death before hospital discharge.

One issue with using this as the response is that the strength of the link between hospital discharge and good health is only moderate at best. Discharge to hospices and inter-hospital transfers are clear examples of the discharge of potentially very ill patients, with such cases requiring explicit action such as removal from consideration when performing ICU benchmarking. Furthermore, the inevitable differences in transfer thresholds (such as with smaller hospitals tending to transfer a greater proportion of patients to hospitals with more advanced care) will introduce a biasing effect that is very difficult to fully account for.

A related issue is that a hospital could effectively play the system with regards to ICU benchmarking by discharging ill patients too early, to ensure that they cannot die before hospital discharge. Although such behaviors seem highly unlikely, and although significant examples should quickly become apparent, the introduction of perverse incentives should remain a concern.

Another issue with death before hospital discharge as a response is that its value will be the same for all ICU stays corresponding to the same hospital stay. This then requires the removal of all but one ICU stay per hospital stay so as to retain sample independence, and forces the inclusion of the admission reduction step described in Section 3.5.2.

For ICU benchmarking, death before hospital discharge has the added issue that it fails to disentangle the quality of an ICU from that of the remainder of the hospital in which the ICU is located. If an ICU patient recovers, they will usually be sent to the hospital floor before hospital discharge. Worse or better than average quality of care provided by the hospital floor
would then result in more or fewer deaths, respectively, with the estimated ICU quality of care unduly reduced or increased as a result.

One might then suggest that the solution is not to use an alternative to death before hospital discharge, but instead to consider hospital benchmarking as opposed to ICU benchmarking. However, using an ICU Database for hospital benchmarking introduces further issues: A patient will tend to spend time in the hospital floor before transfer to the ICU such that a patient who has received better or worse than average care during this time will appear less or more ill than they were on hospital admission, with the hospital’s apparent quality of care being unduly distorted in the opposite direction. To overcome this, one could use a database that contains events spanning a patient’s entire hospital stay, as opposed to the ICU Databases that are the focus of the current work and which contain events that fall almost exclusively within a patient’s ICU stay.

**Death before ICU discharge**

Given the focus of this thesis on ICUs, the most obvious potential response appears to be death before ICU discharge. Changing from death before hospital discharge to death before ICU discharge would both remove the need for an admission reduction step and resolve the benchmarking biases introduced by differences in the quality of care provided by the hospital floor post-ICU discharge. However, ICU discharge tends to be less strongly coupled to patient recovery, with transfers between the ICUs within a single hospital much more common and potentially more dependent on non-health related circumstances, such as bed availability, than transfers between hospitals. ICUs also tend to have equipment that may be unavailable on the hospital floor, such as mechanical ventilators, such that health-related transfers between departments within a single hospital may depend more on a patient’s specific needs rather than their overall state of recovery.
3.3 Response Selection

**Death within 28 days**

Death within 28 days of either ICU admission, or of a risk prediction being made if one is focused on risking, is an alternative response that should be more strongly coupled to patient health. Such a response is not a direct function of patient transfers and is therefore more robust to changes in transfer policy. The main issue with its use in ICU MPMCs is that often ICU Databases do not contain the information required to evaluate death within 28 days for the majority of their patients. This is the case for the APACHE Database (Section 3.7) that is the primary ICU Database used in this thesis. As such, death within 28 days is left as a potential alternative to death before hospital discharge that may be more appropriate when working with ICU Databases that contain the required information.

**Time to death**

Rather than taking death before some fixed point (ICU or hospital discharge) or death within some fixed interval (such as 28 days after ICU admission) as a binary response one might instead use time to death as a numeric response. Such a response might offer additional information about the health of the patients used to train MPMs (with those that died earlier tending to be more unwell at ICU admission than those who died later) such that more accurate MPMs might be created. Using a numeric response could also allow for MPMs that output survival curves as opposed to just survival probabilities; such survival curves could be particularly useful for risking. Consider a hypothetical situation in which there are two patients, one with a 20% probability of death spread uniformly over one hour and the other with an 80% probability of death spread uniformly over one week. Then the second patient would be flagged as the highest risk of the two patients by an alert biased purely on the patients’ probabilities of death while the information in a survival curve might lead to the flagging of the first patient as the one in greater need of an immediate intervention. The
main issue with using time to death as the response for ICU MPMCs is again that many ICU Databases do not contain the required information.

**Death from a given disease**

The responses considered thus far are all proxies for a patient’s overall state of health. However, for various risking applications it could be more useful to have an estimate for the extent to which a patient is suffering from a particular disease. One potential response that could provide such estimates is death from a specific disease before hospital (or ICU) discharge. Often ICD9 or ICD10 codes are recorded for hospital patients which list the main diseases from which the patients are believed to have suffered. For a patient who dies in hospital, the primary ICD9 or ICD10 code usually reports the disease that is thought to be the primary cause of the patient’s death. Thus, one could use the presence of a specific ICD9 or ICD10 code for patients who have died in hospital as an indication of death from a specific disease. Death from a specific disease could then be used to train an MPM that would output the probability of death from the given disease, with this probability then used as a proxy for the degree to which the patient suffers from the disease. Such MPMs might then aid clinicians where there are insufficient evidence-based recommendations built on randomized controlled trials to fully inform decision making [4].

However, the use of ICD9 or IDC10 codes as a response in ICU MPMCs is associated with various downsides. As is the case for the APACHE Database (Section 3.7) many ICU Databases do not include ICD9 or ICD10 codes. The probability of death from a given disease is not necessarily a good measure of disease severity. Some diseases may be entirely debilitating but non-fatal and measuring severity in terms of probability of death from a given disease would ignore the impact of such diseases entirely. Further, because one of the main uses of ICD9 and ICD10 codes recorded in US ICUs is in calculating hospital remunerations, it is feasible that such records may be biased.
Conclusion

Given that death before hospital discharge facilitates comparisons with existing MPMs, and that no alternative considered herein seems definitively superior when considered in conjunction with the constraints imposed by the APACHE Database, death before hospital discharge will be used as the response for the remainder of this thesis.

3.4 Feature Extraction

Most standard machine learning techniques would struggle with a patient profile represented as a list of events as presented in Table 3.2. The present Section focuses on how a patient profile can be converted into a more machine learning ready form known as a ‘design matrix’. Let a ‘feature value’ be a scalar extracted from a patient profile. Let a ‘feature’ be a particular method for extracting a feature value, such as the evaluation of a patient’s mean heart rate or whether a patient was on a mechanical ventilator 24 hours after ICU admission. Then a design matrix is a matrix with columns corresponding to different features and rows corresponding to different patient profiles.

Let a ‘feature extractor’ be a function that takes as input a set of patient profiles and that outputs a design matrix. In this thesis the following feature extractors are considered:

1. The Standard Extractor: The feature extractor used in the Standard MPMC, with details provided in Subsection 3.4.2. The set of features output by the Standard Extractor shall be referred to as the ‘Standard Features’ and are listed in Table 3.4.

2. The Extended Extractor: A feature extractor that aims to autonomously generate a large number of features, with details provided in Subsection 3.4.3.

3. The Reduced Extractor: A version of the Standard Extractor that omits chronic condition flags and diagnosis groups.
4. The Basic Extractor: A feature extractor that outputs only age and the Acute Physiology Score (APS), a measure of physiological derangement detailed in Subsection 3.4.1.

5. The Demographic Extractor: A feature extractor that outputs only the subset of features in Table 3.4 that are of the ‘demographic’ class.

6. The Physiological Extractor: A feature extractor that outputs only the subset of features in Table 3.4 that are of the ‘physiological’ class.

The first and second of these feature extractors represent the standard approach and a potentially more accurate alternative, respectively. The third acts as a bridge between the first two feature extractors that is required because of the limitations of the ICU Databases used in this thesis. The final three feature extractors are included to better represent the full range of feature extractor inducers that might be used in generic MPMCs, and are used in Chapter 6.

Let a ‘feature extractor inducer’ be a function that takes as input a set of patient profiles and that outputs a feature extractor. For each of the feature extractors listed above (excluding the Extended Extractor) the corresponding feature extractor inducer simply outputs the given feature extractor, irrespective of the patient profiles to which it is applied. This is because each of these feature extractors represent the evaluation of a fixed set of features. In contrast, the set of features that make up the Extended Extractor depend on the patient profiles to which the Extended Extractor Inducer is applied.

### 3.4.1 The Acute Physiology Score

The third version of the Acute Physiology Score (APS) was created for the APACHE III Model [47] and combines scores for a variety of types of physiological derangement. For each physiological feature a mapping is defined from feature values onto scores. For each type of ‘event’, the score associated with the ‘worst’ event is output. The APS then sums these scores to output a single number which can then be used as a feature value. The
3.4 Feature Extraction

<table>
<thead>
<tr>
<th>Heart Rate Threshold / bpm</th>
<th>Contribution to APS</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 39</td>
<td>8</td>
</tr>
<tr>
<td>40-49</td>
<td>5</td>
</tr>
<tr>
<td>50-99</td>
<td>0</td>
</tr>
<tr>
<td>100-109</td>
<td>1</td>
</tr>
<tr>
<td>110-119</td>
<td>5</td>
</tr>
<tr>
<td>120-139</td>
<td>7</td>
</tr>
<tr>
<td>140-154</td>
<td>13</td>
</tr>
<tr>
<td>≥ 155</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 3.3 Potential contributions to the APS provided by different heart rate measurements. This table is an extract from Figure 1 of the APACHE III paper [47] and can be used to determine which of multiple heart rate measurements is the ‘worst’ and the associated contribution to the APS that such a worst value provides.

Contributions to the APS provided by different types of physiological event are presented in Figures 1 and 2 of the APACHE III paper [47], with that for the heart rate reproduced in Table 3.3 of the present thesis for illustrative purposes.

As an example, suppose that one has three heart rate measurements of 120, 111 and 105. Then it can be seen from Table 3.3 that the greatest contribution to the APS is from the value of 120. As such the ‘worst’ heart rate would be recorded as 120. Ties are broken by selecting the measurement which would need to be perturbed most in order to be placed into a lower bracket. For example, suppose that one has two heart rate measurements of 41 and 115; either could potentially contribute 5 to the APS. But the value of 41 would have to be shifted by 9 to enter the lower contributing 50-99 bracket, while the value of 115 would only have to be shifted by 6 to enter the lower contributing 100-109 bracket. As such the value of 41 is deemed to be the ‘worst’ value. When all measurements of a given type would contribute zero to the APS, the value which is nearest to a higher contributing bracket is deemed to be the ‘worst’.
3.4 Feature Extraction

3.4.2 The Standard Extractor

The Standard Extractor is the feature extractor used in the Standard MPMC and the Standard Features that it outputs are listed in Table 3.4. Each physiological feature output by the Standard Extractor provides the ‘worst’ recorded value, with what constitutes the ‘worst’ value for each feature determined by its potential contribution to the APS (as set out above). Discussion of how the Standard Features are distributed is delayed until Subsection 3.7.1, once feature preprocessing has been introduced (Section 3.6), so that feature distributions before and after preprocessing can be readily compared.

The Glasgow Coma Score (GCS) is the only Standard Feature that is a composite measurement, as the GCS represents a combination of three different measures of neurological impairment; eye response, verbal response and motor response. Ideally the GCS would be replaced with three features corresponding to each of the three component measures. However, this splitting has not been performed for the Standard Features because the individual measures are not available in the APACHE Database. Another issue with composite measures such as the GCS is that they are sometimes superseded by superior alternatives, with an alternative to the GCS already under consideration [84].

The ‘Standard Extractor Inducer’ is a feature extractor inducer that outputs the Standard Extractor irrespective of the patient profiles to which it is applied. This thesis implements only the feature extractor, rather than the feature extractor inducer, used in the APACHE IV paper [87] because the latter is heuristic and cannot be faithfully reproduced without considerable clinical knowledge. Similar considerations apply to each of the Reduced Extractor Inducer, Basic Extractor Inducer, Demographic Extractor Inducer and Physiological Extractor Inducer.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Demographic</td>
<td>Age of the patient at hospital admission in years.</td>
</tr>
<tr>
<td>Previous Length of Stay</td>
<td>Demographic</td>
<td>Time between hospital admission and ICU admission.</td>
</tr>
<tr>
<td>Previous Location</td>
<td>Demographic</td>
<td>A categorical feature that denotes the location from which the patient was transferred to the ICU. Possible categories are: Direct admission, floor, ICU transfer, operating room, other hospital, recovery room, step down unit and other.</td>
</tr>
<tr>
<td>Chronic Condition Flags</td>
<td>Other</td>
<td>A set of binary flags each indicating the presence of one of: Aids, Cirrhosis, Hepatic Failure, Immunosuppression, Lymphoma, Myeloma, Tumor with metastasis.</td>
</tr>
<tr>
<td>Diagnosis Group</td>
<td>Other</td>
<td>One of 124 conditions deemed to be the primary cause of ICU admission.</td>
</tr>
<tr>
<td>Albumin</td>
<td>Physiological</td>
<td>Abnormal levels can be indicative of liver or kidney disease.</td>
</tr>
<tr>
<td>Bilirubin</td>
<td>Physiological</td>
<td>Abnormal levels can be indicative of liver disease.</td>
</tr>
<tr>
<td>BUN</td>
<td>Physiological</td>
<td>Blood urea nitrogen.</td>
</tr>
<tr>
<td>Creatinine</td>
<td>Physiological</td>
<td>Abnormal levels can be indicative of liver or kidney disease.</td>
</tr>
<tr>
<td>GCS</td>
<td>Physiological</td>
<td>Glasgow coma scale. A measure of patient alertness. A value of 15 (fully alert) was imputed for sedated patients.</td>
</tr>
<tr>
<td>Glucose</td>
<td>Physiological</td>
<td>Concentration of glucose in the blood. High values can be indicative of diabetes.</td>
</tr>
<tr>
<td>Hematocrit</td>
<td>Physiological</td>
<td>The ratio of the volume of red blood cells to the total volume of blood. Low values can be indicative of leukemia.</td>
</tr>
<tr>
<td>Heart Rate</td>
<td>Physiological</td>
<td>Can be indicative of cardiovascular diseases.</td>
</tr>
<tr>
<td>Intubated</td>
<td>Physiological</td>
<td>Binary flag. Set to positive if the patient was intubated.</td>
</tr>
<tr>
<td>MAP</td>
<td>Physiological</td>
<td>Mean arterial blood pressure. Can be indicative of cardiovascular diseases.</td>
</tr>
<tr>
<td>PaCO₂</td>
<td>Physiological</td>
<td>Partial pressure of carbon dioxide in arterial blood. High values can be indicative of respiratory and cardiovascular issues.</td>
</tr>
<tr>
<td>PaFi</td>
<td>Physiological</td>
<td>Ratio of partial pressure of oxygen in arterial blood to fractional inspired oxygen. Can be indicative of respiratory disorders.</td>
</tr>
<tr>
<td>PaO₂</td>
<td>Physiological</td>
<td>Partial pressure of oxygen in arterial blood. Low values can be indicative of respiratory and cardiovascular issues.</td>
</tr>
<tr>
<td>pH</td>
<td>Physiological</td>
<td>Blood acidity. Low values can be indicative of asthma and chest injuries.</td>
</tr>
<tr>
<td>Respiratory Rate</td>
<td>Physiological</td>
<td>Abnormal values can be indicative of respiratory issues.</td>
</tr>
<tr>
<td>Sodium</td>
<td>Physiological</td>
<td>Sodium levels in blood. Low values can be indicative of kidney or liver disease.</td>
</tr>
<tr>
<td>Temperature</td>
<td>Physiological</td>
<td>High values can be indicative of infections.</td>
</tr>
<tr>
<td>Urine Output</td>
<td>Physiological</td>
<td>Volume output during the last 4 hours. Low values can be indicative of kidney disease.</td>
</tr>
<tr>
<td>Ventilation</td>
<td>Physiological</td>
<td>Flag for whether the patient underwent mechanical ventilation.</td>
</tr>
<tr>
<td>WBC</td>
<td>Physiological</td>
<td>White blood cell count. High values can be indicative of leukemia.</td>
</tr>
</tbody>
</table>

Table 3.4 List of the Standard Features, those that are output by the Standard Extractor. Physiological features report the ‘worst’ value recorded between ICU admission and prediction time, where the ‘worst’ value is taken to be that which would provide the greatest contribution to the APS Score.
3.4.3 The Extended Extractor Inducer

The ‘Extended Extractor Inducer’ is a feature extractor inducer that can be applied to a set of patient profiles and responses in order to output a feature extractor capable of generating a design matrix with a large number and diversity of features, without the need for substantial clinical knowledge. When applied to the MIMIC II Database (Section 3.8) the Extended Extractor Inducer outputs an Extended Extractor that, when applied to a set of appropriate patient profiles, outputs a design matrix incorporating approximately 5,000 features.

The Extended Extractor Inducer first considers each unique event type in turn and classifies it as either numeric or categorical, with binary events taken to be categorical. For each type of categorical event the last value that occurred within 24 hours before prediction time is taken for each stay. For each type of numeric event eight feature values are reported per patient profile: the last, mean, median, minimum, maximum, interquartile range, sum\(^8\) and total number of events, each evaluated for events that occurred within 24 hours before prediction time. Any of these feature values can be missing, with the exception of the total number of events. The interquartile range is output as missing for a given stay if fewer than two unique event values of the appropriate type exist, while the remaining six feature values are only output as missing if no event values of the appropriate type exist.

3.5 Stay Exclusions

At this stage one has the tools necessary to convert an ICU Database into a design matrix, with each row corresponding to a different patient profile and each column corresponding to a different feature. However, such a design matrix may contain inappropriate rows that must be removed before machine learning techniques can be applied.

\(^8\)The inclusion of the sum is to deal with cumulative events such as fluid output.
3.5 Stay Exclusions

3.5.1 Unconditional Exclusions

Certain ICU stays are removed because they are missing key information or because they fall outside of the cohort of interest. Two sets of ‘Unconditional Exclusions’ are considered:

1. Basic Exclusions: These exclude stays without recorded ages or that were for patients younger than 16 years old at ICU admission, in addition to stays that lacked either a response, a membership, a known length of stay, or for whom $T_i < t_i$ (admissions for which the patient was discharged from the ICU before their associated prediction time).

2. Standard Exclusions: These are the exclusion criteria used in the Standard MPMC and include all of the Basic Exclusions in addition to excluding stays for which a patient had burns, or was missing an APS 24 hours after admission, or had a hospital length of stay of greater than 365 days, and stays corresponding to patients admitted after transplant operations, except for hepatic and renal transplants. To have received the required APS the patient must have at least one of each of the following measurements recorded: heart rate, respiratory rate and mean arterial pressure.

3.5.2 Conditional Exclusions

Multiple ICU stays may correspond to the same hospital stay. If one is using death before hospital discharge as the response, then all ICU stays associated with the same hospital stay will necessarily have the same value for the response. This then violates the standard independence assumptions on which most machine learning techniques are based and is resolved through the application of one of the following ‘Conditional Exclusions’:

1. Retain only the first stay per patient out of those that have not already been excluded.

2. Retain only a single random stay per patient out of those that have not already been excluded.
3.6 Preprocessing

The option of retaining only the final stay per patient is not included because it would introduce a bias towards stays associated with higher mortality rates. After the application of the reduction criteria there will be only a single ICU stay per patient, such that the terms ‘stay’ and ‘patient’ may be used interchangeably. The term ‘stay’ will only be used to refer to stays that have been retained beyond the application of the reduction criteria.

3.6 Preprocessing

Many machine learning techniques such as logistic regression cannot deal with missing or erroneous values implicitly, and can suffer from the inclusion of features from highly skewed distributions. To ensure that the benchmark of logistic regression does not under-perform, design matrices must be processed to resolve these issues. Preprocessors are introduced to fulfill this role and are taken to be functions that both input and output a design matrix.

A preprocessor inducer takes as input a design matrix and outputs a preprocessor. The Standard Preprocessor Inducer is introduced as a preprocessor inducer that recursively transforms features and median imputes missing values and outliers, as described in Appendix C, in order to generate the ‘Standard Preprocessor’. The Standard Preprocessor Inducer is the only such inducer considered because preprocessing is taken to be a requirement rather than a focus of this thesis.

3.7 The APACHE Database

The APACHE Database is an ICU Database containing the medical records of 231,019 patients admitted between 2002 and 2008 to one of 109 American ICUs. It is an updated version of the database used in the original APACHE IV paper [87], and is perhaps the most famous ICU Database focusing on benchmarking and quality improvement [20].
The version of the APACHE Database available to the author included for each patient: a single value for each feature in Table 3.4, a binary flag specifying whether the patient died before hospital discharge and indices specifying the ICU and hospital to which the patient was assigned. The database had already undergone Event Exclusion by the Deterministic Variable Prediction Time Inducer (Section 3.2.1), the Standard Extractor (Section 3.4.2), Standard Exclusions (Section 3.5.1) and the first set of Conditional Exclusions (Section 3.5.2).

The available form of the APACHE Database precludes its use in investigations into continuous MPMs and alternative feature extraction approaches. However, the relatively large number of ICUs and inclusion of early prediction times make it an excellent database for supporting benchmarking investigations, and it is used as such in the fitting of simulation parameter values in Chapter 6 and in risking performance investigations in Chapter 9.

The number of patients and the mortality rate for each ICU in the APACHE Database are displayed as a funnel plot in Figure 3.1. The dashed lines in this figure represent the limits of a central 95% confidence interval for mortality rates, under the assumption that patient risk is independent of ICU assignment. The fact that the majority of points lie outside this interval is overwhelming evidence against the validity of this assumption, indicating that the ICUs in the APACHE Database contain a substantial amount of case-mix variability and/or quality of care variability.

### 3.7.1 Feature Distributions

Feature distributions before and after the application of the Standard Preprocessor are presented as normal density plots in Figure 3.2. Only the most interesting distributions from the Standard Features have been included, with the remainder providing little additional insight (they are provided in Appendix D). A normal probability plot maps the quantiles of
Fig. 3.1 Number of patients and mortality rates for each of the ICUs in the APACHE Database. The dashed lines give a central 95% confidence interval for mortality rates, under the assumption that patients from different ICUs have the same risk of mortality.

a feature against the equivalent quantiles of a normal distribution. If a feature is normally distributed then its points on a normal probability plot should approximate a straight line.

The top left panel in Figure 3.2 displays how patient temperatures are distributed prior to preprocessing, with the red points representing the values that are discarded by the Standard Preprocessor. The red points in this plot largely fall into three categories; those that are far too high (>50°C), those that are far too low (<18°C) and those that are ‘somewhat low’ (≈30°C). Those that are far too high are clearly examples of recording temperatures in Fahrenheit rather than Centigrade and could potentially be converted back into the appropriate range. The values that are far too low are examples of errors without clear cause and for which there is no obvious means of conversion back into meaningful values. These two sets of extreme values represent genuine outliers that should be removed, although clearly a human could perform this task better than the Standard Preprocessor by converting rather than simply
3.7 The APACHE Database

Fig. 3.2 Normal probability plots for temperature (left panels), white blood cell count (center panels) and Glasgow Coma Scale (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.

discarding the very high values. In contrast, it is unclear whether the ‘somewhat low’ values are correct or if they are underestimates due to some issue such as the inappropriate use of thermometers.

The distribution of points in the bottom left panel of Figure 3.2 is far more linear than that for the top left panel. This indicates that the Standard Preprocessor has succeeded in making patient temperatures far more normally distributed, as intended. It is further noted that the total number of temperature values that were removed by the Standard Preprocessor is only 113 out of the original 183,167, showing that this increase in normality is not due to the discarding of huge numbers of feature values.
3.7 The APACHE Database

The top center panel in Figure 3.2 displays how patient white blood counts are distributed prior to preprocessing. These feature values are far from normally distributed, with some extremely large feature values (in red) that are discarded by the Standard Preprocessor. That the bottom center panel in Figure 3.2 is far more linear than the top center plot indicates that the Standard Preprocessor has again succeed in forcing a feature to be more normally distributed. However, the very large values that have been removed were not necessarily erroneous and their removal is almost certainly throwing out useful information that could potentially be used to diagnose genuine medical conditions (for example leukemia). The trade-off is that leaving in these large values would make the use of the white blood cell count in a linear regression model extremely difficult, with features of this form potentially treated better through the APS-type approach of feature quantization.

The top right plot in Figure 3.2 displays how Glasgow Coma Scale (GCS) is distributed prior to preprocessing. That this plot is identical to the bottom right plot in Figure D.1 (except for the rescaling of the y-axis) and contains no red points indicates that the Standard Preprocessor neither throws out any values nor applies any transformation (other than rescaling) to the GCS. However, this is in spite of the GCS clearly not being normally distributed. Instead the GCS is from a bivariate distribution with a large peak at 15 (representing entirely responsive patients) and a smaller peak at 3 (representing entirely unresponsive patients). The Standard Preprocessor has no mechanism through which to transform such a bivariate distribution into a more normal distributed feature. However, the expectation is that such a bivariate distribution will not damage a logistic regression model nearly as much as a highly-skewed distribution; logistic regression models cope well with the natural extension to a binary feature.

The distributions of the remaining 17 numeric features of the Standard Feature Set exhibit similar but less extreme versions of the behavior described above for temperature, white blood count and GCS. Generally the Standard Preprocessor Inducer appears to be sufficient
for its role in this thesis, although it is quite clear that a superior preprocessor could be created.

### 3.8 The MIMIC II Database

Version 2.6 of the Multi-parameter Intelligent Monitoring for Intensive Care (MIMIC II) Database contains the medical records of approximately 32,000 ICU patients admitted to the ICUs of the Beth Israel Deaconess Medical Center between September 2001 and September 2008 [18, 69, 70]. For each hospital admission, MIMIC II contains a record of when the patient died if they did so before September 2008, as well as numerous types of physiological measurements and treatments (such as vital signs, lab results and medications) recorded as low frequency time series (generally less frequent than every 15 minutes). For a subset of its patients, the MIMIC II Database also contains a variety of high frequency (many times per second) physiological time series, although these are not considered further in this thesis.

The MIMIC II Database contains only a small number of ICUs from a single hospital, making it inappropriate for investigating ICU benchmarking. It also lacks the admission diagnoses required to evaluate APACHE IV scores, precluding the direct application of the Standard MPMC to it. Furthermore, the admission dates for patients in the MIMIC II Database have been randomly altered for the purpose of patient de-identification, such that changes in patient populations and treatments cannot be taken into account using the available version. However, the MIMIC II Database can be used to investigate continuous models and to help compare the performance of MPMs trained using a large number of patients against MPMs trained using a large number of features, with the MIMIC II Database used in Chapter 9 to help perform each of these investigations.
Chapter 4

Machine Learning

In this thesis, a ‘learner’ is taken to be a method for converting a design matrix into a set of mortality predictions, while a ‘learner inducer’ is taken to be a method for converting a design matrix and a response vector into a learner. An MPM is expressible as a combination of a feature extractor, a preprocessor and a learner. The present Chapter is dedicated to describing learners, due to their central position within MPMs, and of learner inducers within MPMCs, in addition to their considerable complexity. However, it is noted that a full appreciation of the learners and learner inducers presented in this Chapter is not essential for following, or interpreting the findings of, the remainder of this thesis.

A machine learning technique can be viewed as a superset of learner inducers, with each learner inducer within a given machine learning technique differing in terms of its associated hyper-parameter values. For example, two logistic regression models with different degrees of regularization would provide two different learner inducers that nevertheless both belong to the same class of machine learning technique; namely regularized logistic regression. For convenience, a complete list of the learner inducers considered in this thesis is provided in Table 4.1, with the descriptions of the machine learning techniques to which they belong provided throughout the remainder of this Chapter.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Name</th>
<th>Machine Learning Technique</th>
<th>Section</th>
<th>Hyper-Parameter Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>The NULL Model</td>
<td></td>
<td>4.1</td>
<td>-</td>
</tr>
<tr>
<td>NN(^1)</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 1)</td>
</tr>
<tr>
<td>NN(^3)</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 3)</td>
</tr>
<tr>
<td>NN(^9)</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 9)</td>
</tr>
<tr>
<td>NN(^{27})</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 27)</td>
</tr>
<tr>
<td>NN(^{81})</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 81)</td>
</tr>
<tr>
<td>NN(^{243})</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 243)</td>
</tr>
<tr>
<td>NN(^{729})</td>
<td>Nearest-Neighbors</td>
<td></td>
<td>4.2</td>
<td>(\kappa = 729)</td>
</tr>
<tr>
<td>GC</td>
<td>Gaussian Classifier</td>
<td></td>
<td>4.3</td>
<td>-</td>
</tr>
<tr>
<td>RLR(^0)</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 0)</td>
</tr>
<tr>
<td>RLR(^5)</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 5)</td>
</tr>
<tr>
<td>RLR(^{10})</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 10)</td>
</tr>
<tr>
<td>RLR(^{20})</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 20)</td>
</tr>
<tr>
<td>RLR(^{40})</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 40)</td>
</tr>
<tr>
<td>RLR(^{80})</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 80)</td>
</tr>
<tr>
<td>RLR(^{160})</td>
<td>Regularized Logistic Regression</td>
<td></td>
<td>4.4</td>
<td>(\lambda = 160)</td>
</tr>
<tr>
<td>APACH</td>
<td>The APACHE IV Inducer</td>
<td></td>
<td>4.5</td>
<td>-</td>
</tr>
<tr>
<td>CART</td>
<td>Classification and Regression Trees</td>
<td></td>
<td>4.6</td>
<td>-</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forests</td>
<td></td>
<td>4.7</td>
<td>-</td>
</tr>
<tr>
<td>BART</td>
<td>Bayesian Additive Regression Trees</td>
<td></td>
<td>4.8</td>
<td>-</td>
</tr>
<tr>
<td>BEAST</td>
<td>The Bayesian Ensemble of Sigmoidal Trees</td>
<td></td>
<td>4.9</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.1 Table listing all of the learner inducers considered in this thesis. The third column gives the Section that describes the machine learning technique to which the given learner inducer belongs. The hyper-parameters \(\kappa\) and \(\lambda\) represent the number of nearest neighbors and the regulation coefficient, respectively.
Notation and Examples

In this Chapter the notation \( y_i \) and \( x_{ij} \) is used to denote the response and value of the \( j \)’th feature, respectively, for the \( i \)’th patient. The notation \( x_i \) shall be used to denote the vector of values specified by the \( i \)’th row of \( x \). Throughout this Chapter let \( \tau \) denote the set of indices of patients in some training set on which a learner is to be trained. Then a learner inducer will take as inputs the set of features, \( x_{\tau} \), and responses, \( y_{\tau} \), and will output a learner, \( l \).

For illustrative purposes two examples of \( \tau \) are considered in this Chapter: one containing 100 patients and one containing 10,000 patients, with each patient drawn randomly from the APACHE Database. In both cases the design matrix is set to contain only two well-known and predictive features; age and the Glasgow Coma Scale (GCS). The inclusion of only two such features is so that the learners can be easily visualized in terms of their mapping of feature values onto risk estimates.

The distribution of features and responses for the two versions of \( \tau \) are provided in Figure 4.1. Notably the GCS values have had jitter\(^1\) applied to them so as to avoid ties\(^2\) and aid with presentation. Beyond the application of jitter, age and GCS values are presented without preprocessing so as to aid with their interpretation. The strongest trend visible in these plots is that patients with a lower GCS tend to be more strongly associated with a positive response, while to a much lesser extent older patients tend to be more strongly associated with a positive response than younger patients. The expectation then is that learner inducers applied to these design matrices will output learners that predict that old patients with a low GCS are more likely to die than young patients with a high GCS.

\(^1\)The application of jitter involves adding a random number to each entry in the design matrix, drawn from a uniform distribution from \([-\varepsilon, +\varepsilon]\), where \( \varepsilon \) is small compared to the differences in the values for the given feature.

\(^2\)Ties can affect some learner inducers such as Nearest Neighbors and are rare in datasets that contain a moderate or large number of features.
4.1 The Null Inducer (NULL)

The ‘Null Inducer’ is a learner inducer that outputs a learner (the ‘Null Model’) that predicts the risk of all new patients to be equal to the mean survival rate of the patients in the training set:

$$\hat{p}_i = E_{\theta \in \tau}(y_\theta).$$

The Null Inducer has no hyper-parameters and the Null Model has only a single parameter - the estimated probability assigned to all new patients. The Null Model is the simplest possible unbiased learner and both the Null Model and the Null Inducer are independent of their input design matrices. The Null Model cannot account for case-mix or track a patient’s deterioration and therefore cannot be used for either risking or benchmarking. However, the Null Model serves as a useful baseline which all viable Learner Inducers should be expected to exceed.
4.2 Nearest-Neighbors (NN)

A learner trained using $\kappa$-Nearest-Neighbors sets the estimated risk of a new patient to be equal to the mean response shown by the $\kappa$ patients in the training set with feature values most similar to those of the new patient. Nearest-Neighbors is a highly intuitive learner inducer that mimics a simple thought process: ‘patient 10 looks most like patient 5 and the outcome for patient 5 was death before hospital discharge, therefore I expect the outcome for patient 10 to also be death before hospital discharge’.

One key component of Nearest-Neighbors is the approach used to assess which patients are the most similar to a new patient. The most common method is to look for the set of patients with the minimum Euclidean distance between their sets of features and those of the new patient. To be effective, this method requires all types of features to be on similar scales and requires that no features are drawn from highly skewed distributions. Fortunately, the design matrix passed to any Nearest-Neighbor based learner used in this thesis will have already undergone preprocessing (Section 3.6) such that these required criteria are always met. Given this and its general popularity, the Euclidean distance is used from here on as the measure of distance between sets of features within Nearest-Neighbors.

Given a new patient with index $i$, a learner output using Nearest-Neighbors first finds the Euclidean distance, $||x_i - x_\theta||$, between the feature values of the new patient and that of each patient in the training set, $\theta \in \tau$. For some positive integer, $\kappa$, let $\psi^\kappa_i$ be the subset of $\kappa$ patients from $\tau$ that have the smallest values of $||x_i - x_\theta||$. Then the $\kappa$-Nearest-Neighbors learner provides risk estimates of the form:

$$\hat{p}_i = \frac{1}{\kappa} \sum_{\theta \in \psi^\kappa_i} y_\theta. \quad (4.2)$$

As the value of $\kappa$ is increased, the predictions output by the $\kappa$-Nearest-Neighbors learner will become more stable but also more biased. For small values of $\kappa$, Equation 4.2 will
also be prone to outputting probabilities with values 0 or 1. Such estimates are undesirable because they suggest that the learner is entirely certain of a given outcome even with only minimal evidence on which to found such a claim. Taking such overconfident estimates at face value could lead to poor decision-making, and may also introduce infinite values when assessed by various measures of MPM performance, such as the log-likelihood. To resolve these issues, the outputs of $\kappa$-Nearest-Neighbors used in this thesis are mixed with a small portion of the outputs from the NULL Model as:

$$\hat{p}_i = \frac{1}{\kappa + 1} \sum_{\theta \in \psi_i^\kappa} y_\theta + \frac{1}{\kappa + 1} E_{\theta \in \tau}(y_\theta).$$

The risk estimates output by $\kappa$-NN as a function of a patient’s age and GCS for each combination of example training set and for $\kappa = 1$ and $\kappa = 99$ are presented in Figure 4.2. The bottom left panel is almost identical to that output by the Null Inducer because the estimate is always an average over almost all patients from the training set. The remaining three panels all exhibit the desired behavior of their associated learners tending to output greater risk estimates for older patients with smaller GCSs. However, the top two panels exhibit the undesirable property that very small changes in feature values can result in large changes in estimated risks. This is a natural consequence of 1-NN and only becomes more pronounced as the size of the training set increases, as can be seen on moving from the top left to top right panel. However, this issue can be overcome by increasing the number of nearest neighbors, as can be seen on moving from the top right to the bottom right panel. Of the four plots in Figure 4.2, only that associated with 99-NN on 10,000 patients (bottom right panel) could be seen as viable by a clinician. The learners associated with the top two panels would output low risk estimates for a 95-year-old patient with a GCS of 4, an output that is clearly unacceptable, while the bottom left panel could not provide useful predictions at all.

Selecting a good value for the number of nearest neighbors, $\kappa$, is difficult a priori. The only standard value of $\kappa = 1$ was previously shown to result in an unacceptable learner.
Fig. 4.2 Plots for the predicted risk output by $\kappa$-NN as a function of age and GCS for $\kappa = 1$ (top) and $\kappa = 99$ (bottom). The color at a given location represents the predicted risk output by the learner for a patient with the associated age and GCS. The left and right plots are for learners trained on 100 and 10,000 patients, respectively.
Instead a variety of values for $\kappa$ are considered as:

$$\kappa \in \{1, 3, 9, 27, 81, 243, 729\}. \quad (4.4)$$

The Nearest-Neighbors models of this thesis are trained using the ‘class’ package [78] that is a part of the CRAN repository [37] for R [76], a language and environment for statistical computing and graphics.

### 4.3 Gaussian Classifiers (GC)

A Gaussian Classifier assumes that the feature vectors associated with survivors are drawn from one multivariate normal distribution, while the feature vectors associated with non-survivors are drawn from a different multivariate normal distribution. The two multivariate normal distributions are fit using the patients from the training set, and the probability output for a new patient is set to be equal to posterior probability that the patient was drawn from the multivariate normal distribution associated with non-survivors.

Let $MVN(\alpha; \mu, \sigma)$ be the probability density function for a multivariate normal distribution with vector of means $\mu$ and covariance matrix $\sigma$ evaluated at $\alpha$. Further let $MVN(\alpha; \mu_0, \sigma_0)$ and $MVN(\alpha; \mu_1, \sigma_1)$ correspond to the probability distributions associated with survivors and non-survivors, respectively, for all possible values of $\alpha$. Then a Gaussian Classifier takes the form:

$$\hat{p}_i = \frac{\beta MVN(x_i; \mu_1, \sigma_1)}{(1 - \beta) MVN(x_i; \mu_0, \sigma_0) + \beta MVN(x_i; \mu_1, \sigma_1)}, \quad (4.5)$$

The exponential increase in the considered values of $\kappa$ has been chosen because $\kappa$ is a scale parameter; for a scale parameter, an uninformative prior has the property that the probability of the parameter falling in the interval between $\alpha$ and $2\alpha$ is constant for all positive values of $\alpha$, such that one must consider values that increase exponentially in order to avoid introducing a sampling bias when looking to optimize such a scale parameter. The use of tripling rather than less substantial increases for the value of $\kappa$ are to reduce the computational load, as Nearest-Neighbors can be intensive when applied to training and testing sets that each contain large numbers of patients. However, the load associated with Nearest-Neighbors is not so great as to make a hyper-parameter search computationally intractable.
where $\beta \in [0, 1]$, $\mu_0$, $\sigma_0$, $\mu_1$ and $\sigma_1$ are set by the Gaussian Classifier Inducer so as to maximize the likelihood of the $y_t$ given $x_t$.

The risk estimates output by Gaussian Classifiers as a function of a patient’s age and GCS and trained on the example training sets are presented in Figure 4.3. Both plots exhibit the desired behavior of their associated learners tending to output greater risk estimates for older patients with smaller GCSs. Both plots also exhibit the desirable behavior that very small changes in feature values cannot result in large changes in estimated risks. However, the left plot indicates that the outputs from Gaussian Classifiers can be overly certain, potentially due to the lack of regularization in the provided implementation. For example, the left plot falsely suggests that all patients with a GCS of 3 are almost certain to die before hospital discharge. The same issue is also present in the right plot, although to a lesser extent and potentially due more to the model assumptions being invalid rather than due to the lack of regularization. One might also expect the over-fitting exhibited by Gaussian Classifiers to be exacerbated further on transitioning to training sets that contain a large number of features, because the number of parameters in the stated implementation of Gaussian Classifiers increases with the square of the number of features, while for many learner inducers such as logistic regression the relationship is only linear.

### 4.4 Regularized Logistic Regression (RLR)

Logistic regression is a learner inducer that outputs a ‘logistic regression model’. Linear regression assumes that each patient has some numerical response, with an expectation that is given by a weighted sum of the entries in the patient’s feature vector:

$$\hat{y}_i = \beta_0 + \sum_{j=1}^{J} \beta_{ij}x_{ij},$$ (4.6)
Fig. 4.3 Plots for the predicted risk output by Gaussian Classifiers as a function of age and GCS. The color at a given location represents the predicted risk output by the learner for a patient with the associated age and GCS. The left and right plots are for learners trained on 100 and 10,000 patients, respectively.

Such a linear sum is able to take values throughout the real line and is therefore only appropriate for predicting a numeric response. Logistic regression maps probabilities onto the real line using the logistic function, such that the linear regression model can be applied to estimate probabilities for a binary response as:

$$\log \left( \frac{\hat{p}_i}{1 - \hat{p}_i} \right) = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij},$$

where the parameter values, \( \{\beta_0, \beta_1, \ldots, \beta_J\} \), are chosen to maximize the likelihood of \( y_\mathbf{\theta} \) given \( \mathbf{x}_\mathbf{\theta} \) as:

$$\arg\max_\beta \left( \sum_{\theta \in \mathbf{\theta}} \left[ y_\theta \left( \beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j} \right) - \log \left( 1 + e^{\beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j}} \right) \right] \right).$$

Regularization adds a penalty term so that smaller values of \( \beta_j \), for \( j > 0 \), tend to be favored over larger values. Regularization helps to increase the stability of a logistic
4.4 Regularized Logistic Regression (RLR)

Regression model’s predictions and is included in this thesis to help ensure that, if alternative methods outperform logistic regression, it is due to their strength as opposed to a weakness in the implementation of logistic regression. For a given hyper-parameter value, $\lambda \in [0, \infty)$, a mixture of $L_1$ and $L_2$ regression is used and the likelihood maximization is modified to become:

$$\arg\max_{\beta} \left( \sum_{\theta \in \mathbf{t}} y_{\theta} (\beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j}) - \log \left( 1 + e^{\beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j}} \right) \right) - \lambda \sum_{j=1}^{J} (|\beta_j| + \beta_j^2).$$

(4.9)

A further potential issue with logistic regression is that for most physiological features (such as heart rate and blood pressure), intermediate values tend to indicate good health while extreme values tend to indicate poor health. To allow the regularized logistic regression (RLR) models trained in this thesis to cope with such trends, it is assumed that all such models have square terms included, with Equations 4.7 and 4.9 modified to become:

$$\log \left( \frac{\hat{p}_i}{1 - \hat{p}_i} \right) = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} + \sum_{j=J+1}^{2J} \beta_j x_{ij}^2$$

(4.10)

and

$$\arg\max_{\beta} \left( \sum_{\theta \in \mathbf{t}} y_{\theta} (\beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j} + \sum_{j=J+1}^{2J} \beta_j x_{\theta j}^2) - \log \left( 1 + e^{\beta_0 + \sum_{j=1}^{J} \beta_j x_{\theta j} + \sum_{j=J+1}^{2J} \beta_j x_{\theta j}^2} \right) \right) - \lambda \sum_{j=1}^{2J} (|\beta_j| + \beta_j^2)).$$

(4.11)

The RLR models of this thesis are trained using the ‘penalized’ package that forms part of the CRAN R library. The approach is based around gradient descent and combines efficient methods for both L1 and L2 optimization [30, 25, 77]. In gradient descent the negative log of the likelihood is differentiated with respect to the model parameters ($\beta$) and the ‘gradient of fastest descent’ identified as the unit vector (in the parameter space) associated with the greatest decrease in the negative log-likelihood. Gradient descent then begins by spawning an initial set of parameter values that are iteratively updated by moving down and then re-evaluating the gradient of fastest descent.
The risk estimates output by learners trained using each of logistic regression and regularized logistic regression (with $\lambda = 5$), as a function of a patient’s age and GCS, are presented in Figure 4.4. The two right panels are very similar and exhibit the desired behaviors that their associated learners tend to output greater risk estimates for older patients with smaller GCSs and that very small changes in feature values cannot result in large changes in estimated risks. The similarity is to be expected, as the effects of regularization become less pronounced as the number of patients increases, while the smoothness is a natural property of logistic regression models. In these two panels a small quadratic term is visible wherein the gradient of lines of constant estimated risk increases as age increases. In the top left panel this quadratic term is reversed, indicating that the unregularized learner inducer has accidentally modeled an effect that does not actually exist. The bottom left panel, which incorporates regularization, has implemented a much more conservative model that only considers GCS. Relative to the regularized model (top left panel), the incorrect quadratic age term has been removed and so too has the correct linear age term. It is likely that both of the left panels would be rejected by clinicians as their corresponding learners would state that a 70-year-old with a GCS of 4 has the same risk as a 100-year-old with a GCS of 4, although the top left panel would likely be considered more acceptable than the bottom left panel.

Two standard methods for selecting the regularization constant, $\lambda$, are provided by Akaike’s Information Criterion [3] and the Bayesian Information Criterion [72]. However, the basis for these methods is founded more in model selection than in predictive accuracy optimization. Instead, selecting the optimal value of $\lambda$ from a range of possible values seems more appropriate, with the following values for $\lambda$ considered:

$$\lambda \in \{5, 10, 20, 40, 80, 160\}.$$  \hspace{1cm} (4.12)
4.4 Regularized Logistic Regression (RLR)

Fig. 4.4 Plots for the predicted risk output by logistic regression models as a function of age and GCS, trained using regularized logistic regression with $\lambda = 0$ (top) and $\lambda = 5$ (bottom). The color at a given location represents the predicted risk output by the learner for a patient with the associated age and GCS. The left and right plots are for learners trained on 100 and 10,000 patients, respectively.
4.5 The APACHE IV Inducer (APACH)

The learner inducer used to generate the APACHE IV Model can be viewed as an application of logistic regression with additional preprocessing. Instead of using square terms, the majority of physiological variables are combined in a non-linear manner to form the APS (Subsection 3.4.1), while the non-linearity of the remaining features is accounted for through the incorporation of natural cubic spline terms. A cubic spline is a piecewise function constructed out of third-order polynomials. The ‘knots’ associated with a spline define the limits associated with these polynomials. A cubic spline is constrained to be continuous across knots, as are its first and second derivatives. A natural cubic spline is further constrained to have second and third derivatives set to zero beyond the outer knots.

The learner inducer used to generate the APACHE IV Model allowed each of age, previous length of stay and APS to be independently modeled as a natural cubic splines with 5, 4 and 5 terms, respectively. The locations of the knots associated with these splines were set by the inducer used to generate the APACHE IV Model as 27, 51, 64, 74 and 86 for age, 3 hrs, 10 hrs, 19 hrs and 2.8 days for previous length of stay and 10, 22, 32, 48 and 89 for the APS.

The APACH inducer replicates the inducer used to generate the APACHE IV Model. The number and location of knots output by APACH are set to be identical to those in the APACHE IV Model. However, an application of APACH regenerates all other parameter values, such as regression weightings and spline coefficients, by maximizing the likelihood of the $y_\tau$ given the $x_\tau$. APACH can only be applied to the features output by the Standard Extractor (Section 3.4) and it is anticipated that APACH and RLR will output learners that offer similar performance.
4.6 Classification and Regression Trees (CART)

Classification and regression trees are learners that partition the feature space into hyper-rectangles. Classification trees are extremely easy to interpret because they output a value that only depends on the answer to a set of yes/no questions. CART [12] is a learner inducer that outputs either a classification or regression tree, with consideration here limited to classification trees because hospital mortality is a binary response.

In CART the search for an appropriate binary partition is achieved through a greedy top down splitting algorithm. Let \( I \) be the identify function, with \( I(\alpha) \) taking the value 1 if \( \alpha \) is true and 0 otherwise for an arbitrary input \( \alpha \). Suppose that one has a hyper-rectangle, \( \Gamma_0 \), that may be partitioned into two hyper-rectangles, \( \Gamma_1(j, \beta) \) and \( \Gamma_2(j, \beta) \), generated by splitting along the hyperplane \( x_j = \beta \). Further suppose that \( \Theta(\Gamma) \) outputs the Gini impurity associated with \( \Gamma \) as:

\[
\Theta(\Gamma) = 2 \left( \sum_{\theta \in \tau} (y_\theta I(x_\theta \in \Gamma)) \right) \left( \sum_{\theta \in \tau} (I(x_\theta \in \Gamma)) - \sum_{\theta \in \tau} (y_\theta I(x_\theta \in \Gamma)) \right).
\]  

Then the values of \( j \) and \( \beta \) are chosen as:

\[
\argmin_{j, \beta} (\Theta(\Gamma_1(j, \beta)) + \Theta(\Gamma_2(j, \beta))).
\]

Such a split is then accepted if its associated decrease in Gini impurity is greater than some threshold \( \rho \):

\[
\argmin_{j, \beta} (\Theta(\Gamma_1(j, \beta)) + \Theta(\Gamma_2(j, \beta))) - \Theta(\Gamma_0) \geq \rho.
\]

Splitting continues until no further splits are available that can decrease the Gini impurity by more than \( \rho \). This provides a partitioning of the feature space into \( \Phi \) hyper-rectangles,
\( \{\Gamma_1, \Gamma_2, \ldots, \Gamma_\Phi\} \). The tree associated with such a partition then makes estimates of the form:

\[
\hat{p}_i = \sum_{\phi=1}^{\Phi} \frac{\sum_{\theta \in \tau} \left( y_\theta \bar{I}(x_\theta, \in \Gamma_\phi) \right)}{\sum_{\theta \in \tau} \left( \bar{I}(x_\theta, \in \Gamma_\phi) \right)} \bar{I}(x_i, \in \Gamma_\phi).
\]  

(4.16)

Usually the value of \( \rho \) would be selected via a hyper-parameter sweep, as this value can greatly affect predictions and classification trees are extremely fast to train. However, in this thesis classification trees are exclusively used to support simulations and within this setting the requirement is that the learner inducer is able to provide learners that vary greatly, even when receiving very similar training sets (as discussed in Subsection 6.1.2). Because of this all of the classification trees trained in this thesis use \( \rho = 0.001 \), a small value that will give rise to diverse although somewhat inaccurate learners. The alternative of performing a hyper-parameter sweep is avoided because although it would give more accurate learners, they would also be less diverse, and would therefore not meet the simulation requirements.

Examples of risk estimates output by classification trees trained using CART are presented in Figure 4.5. These two plots exhibit similar behavior to those for 1-NN, with the undesirable property that very small changes in feature values can result in large changes in estimated risks even in the presence of very large training sets. The division of the parameter space into rectangles is immediately apparent and makes classification trees easy to apply. However, these trees would likely be rejected by a clinician because of their undesirable behavior.

Classification trees tend to be highly unstable relative to small changes in the patients included in the training set, with this instability tending to cause individual classification trees to provide poor predictive accuracy. However, this issue can be overcome by averaging over a large number of classification trees that have been trained using different subsets of the training patients, as is achieved through the application of Random Forests (below). The classification trees of this thesis are trained using the ‘rpart’ package [12] that forms part of the CRAN R library.
4.7 Random Forests (RF)

Random Forests (RF) is a learner inducer that outputs a Random Forest: a learner that combines the outputs from numerous classification trees [11]. Random Forests have two hyper-parameters: the total number of trees, $\beta$, and the number of features considered during each split, $\delta$. For a single Random Forest, a total of $\beta$ trees are grown using CART, as described in Section 4.6, except that:

- The minimum increase in the Gini impurity, $\rho$, is set to be equal to zero.
- At each split only a random subset of $\delta$ features are considered.

The values for $\beta$ and $\delta$ used in this thesis are set to their default values of $\beta = 500$ and $\delta = \sqrt{J}$, where $J$ is the total number of columns in the design matrix. The value of $\delta$ is set to this default rather than being left as a tunable hyper-parameter because of the considerable computational time associated with the training of Random Forests on large design matrices.
with sizable values of $\delta$. The fixing of $\delta$ is viable because its default value tends to be near optimal [7]. In terms of predictive accuracy, the value of $\beta$ should be as large as possible, with a values exceeding 200 generally considered to be sufficient [34].

The risk estimate output by a Random Forest for the $i$'th patient is set to be the mean of the values output for that patient by its classification trees. The ensemble approach of Random Forests tends to provide far more stable and accurate predictions than individual classification trees, although at the expense of increased computational complexity and loss of interpretability.

Examples of risk estimates output by Random Forests are presented in Figure 4.6. Both plots exhibit the desired behavior of their associated learners tending to output greater risk estimates for older patients with smaller GCSs. However, both plots also to a moderate extent display the undesirable property that very small changes in feature values can result in large changes in estimated risks. Overall, the degree of over-fitting present in the outputs from a Random Forest may make them clinically unpalatable, even if they do tend to offer a high degree of predictive accuracy. The Random Forests of this thesis are applied using the ‘randomForest’ package that is a part of the CRAN R library.

### 4.8 Bayesian Additive Regression Trees (BART)

Bayesian Additive Regression Trees (BART) is another example of an ensemble learner made up of multiple classification trees [16]. BART is designed to promote the inclusion of weak learners, with such trees favored through the application of appropriate priors. BART benefits from both strong predictive accuracy and a Bayesian framework that allows for the direct evaluation of confidence intervals and conditional probabilities. The details required to implement BART are substantial and fall outside of the scope of the current work; they are available in the original publication of Chipman et. al. [16].
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST) is another example of a learner inducer that creates and updates an ensemble of weak learners. The BEAST differs from BART in terms of its updating approach and the form that its weak learners take,
The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST) with the BEAST relying on the Metropolis-Hastings algorithm for updates, and contains weak learners that consist of products of sigmoid functions. For each patient, the final prediction output by the BEAST is given by the mean over the predictions output by each learner, averaged over many iterations. The benefits of the BEAST include high overall performance and implicit handling of missing values and outliers - as are prevalent in ICU Databases. The main drawback of the BEAST is that it is slow to train to such an extent that it cannot be guaranteed to converge within a reasonable time-frame when applied to a sizable design matrix.

The presentation of the BEAST provided in this Section is split into four parts. The first (Subsection 4.9.1) details the structure of the forest of trees that the BEAST maintains. The second (Subsection 4.9.2) specifies how the forest is initialized. The third (Subsection 4.9.3) describes how the forest is updated. Finally, the fourth (Subsection 4.9.4) details how the BEAST uses its forest to make predictions. The definitions of variables and functions
introduced in this Section using capital Greek letters are taken to persist throughout the entire Section.

4.9.1 Forest Structure

A ‘split’ is taken to be a mapping from a set of patient features onto a real value and can be specified as a length 4 vector, \( (\Theta_1, \Theta_2, \Theta_3, \Theta_4) \), where \( \Theta_1 \in \{1, 2, \ldots, K\} \), \( \Theta_2 \in \mathbb{R} \), \( \Theta_3 \in \mathbb{R} \) and \( \Theta_4 \in \mathbb{R} \). The output of the \( \psi \)th split, \( \delta_\psi \), applied to the features of the \( i \)th patient is denoted \( \delta_\psi(x_i) \). Let \( \xi \) define a function of the form:

\[
\xi(\eta_1, \eta_2, \eta_3) = 1/(1 + e^{-\eta_1(\eta_2 - \eta_3)}),
\]

for arbitrary values \( \eta_1 \in \mathbb{R}, \eta_2 \in \mathbb{R} \) and \( \eta_3 \in \mathbb{R} \). Then the application of \( \delta_\psi \) to the features associated with the \( i \)th patient is defined to yield a real value as:

\[
\delta_\psi(x_i) = \frac{\xi(\Theta_2, x_{i\Theta_1}, \Theta_3) - \xi(\Theta_2, 0, 0)}{|\xi(\Theta_2, 1, 0) - \xi(\Theta_2, -1, 0)|} + \xi(|\Theta_2|, 1, 0) - 1/2
\]

where the value of \( x_{i\Theta_1} \) in Equation 4.18 is replaced with \( \Theta_4 \) if the value of \( x_{i\Theta_1} \) is missing.

Example splits are presented in Figure 4.8. The left two panels are for splits with large \( \Theta_2 \) values, which give rise to step functions. Such splits can be used to produce decision trees. The top and bottom left panels have different \( \Theta_3 \) values which result in a shifting of the decision boundary. The two central panels are for splits with small \( \Theta_2 \) values, which give rise to linear models. The right two panels are for splits with moderate \( \Theta_2 \) values and differ in terms of the sign of \( \Theta_2 \), the changing of which is equivalent to a reflection about \( x = \Theta_3 \).

A ‘tree’ is taken to be a mapping from a set of features onto a real value, with the output of the \( \Psi \)th tree, \( \tau_\Psi \), applied to the features of the \( i \)th patient denoted \( \tau_\Psi(x_i) \). The \( \Psi \)th tree is taken to be composed of \( \nu \) splits and one additional real value: \( (\delta_1, \delta_2, \ldots, \delta_\nu, \Upsilon) \). Applying \( \tau_\Psi \) to the features associated with the \( i \)th patient is taken to yield a real value
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

Fig. 4.8 Six possible splits defined for the BEAST. Each panel provides the value of the given split (y-axis) evaluated at the given value of some arbitrary feature (x-axis) that is assumed to have pre-mapped onto the quantiles of a standard normal distribution. The six panel differ in terms of the $\Theta_2$ and $\Theta_3$ values associated with each split that they represent.

according to the following function:

$$\tau_\psi(x_i) = \gamma \prod_{\psi=1}^V \delta_\psi(x_i)$$  \hspace{1cm} (4.19)

Example trees are presented in Figure 4.9. The left two panels are trees each composed of a single split, with the top left and bottom left panels providing a linear model and binary split, respectively. The central and top right panels are for trees each composed of two splits, with the bottom central and top left panels providing a quadratic and rectangular function, respectively. The bottom right panel is for a tree composed of three splits, that yields a
Fig. 4.9 Six possible trees defined for the BEAST. Each panel provides the value of the given tree (y-axis) evaluated at the given value of some arbitrary feature (x-axis) that is assumed to have pre-mapped onto the quantiles of a standard normal distribution. The title of each panel provides the pairs of $\Theta_2$ and $\Theta_3$ values associated with the given tree’s splits. Each presented tree has been assigned the value $\Upsilon = 1$ for simplicity.

sawtooth function. Each panel in Figure 4.9 is for trees composed of one or more splits each applied to the same feature. However, trees are also able to model interaction terms by combining splits that are each applied to different features.

A ‘forest’ is taken to be a mapping from a set of features onto a probability, and that is composed of $\bar{\Psi}$ trees and two additional values: $(\tau_1, \tau_2, \ldots, \tau_{\bar{\Psi}}, \Gamma_0, \Gamma_1)$, where $\Gamma_0 \in \mathbb{R}$ and $\Gamma_1 \in \mathbb{R}^+$. Applying a forest to the features associated with the $i$’th patient is taken to yield a
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

4.9.1 Probability

The probability according to the following equation:

\[ P(x_i) = \frac{1}{1 + e^{-\left( \Gamma_0 + \sum_{\Psi=1}^{\Psi} \tau_\Psi(x_i) \right)}}. \]  \hspace{1cm} (4.20)

Examples of risk estimates output by a single forest from the BEAST are presented in Figure 4.10.

4.9.2 Initialization

The BEAST is associated with four hyper-parameters: the total number of iterations that are to be performed, \( \bar{\Theta} \), the rate at which forests are to be sampled, \( \bar{\Gamma} \), the maximum tree depth, \( \Delta \), and the total number of trees per forest, \( \bar{\Psi} \). For every BEAST trained in this thesis, these four values are set as \( \bar{\Theta} = 160,000 \), \( \bar{\Gamma} = 2,000 \), \( \Delta = 3 \) and \( \bar{\Psi} = 500 \). The value of \( \bar{\Theta} \) was chosen to be as large as possible without requiring excessive computation time. The value of
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

The value of $\bar{\Psi}$ was set to 500 so as to mirror the equivalent default value used in Random Forests. The value of $\Delta$ was set to 3 to allow the model to cope with second order interaction terms.

The BEAST begins by preprocessing the patient features by splitting categorical features that have more than two levels into binary features. Each feature is then mapped onto the quantiles of a standard normal distribution.\(^4\)

Let $\Lambda(\beta, \gamma)$ be a function that takes as input an integer, $\beta$, and a vector of positive real values, $\gamma$, of length $\beta$, and that outputs a single value from $\{1, 2, \ldots, \beta\}$, where the probability of the $\beta$'th value being selected is proportional to the $\beta$'th element of $\gamma$. Let $\tilde{\Delta}$ be a function that generates a new split with $\Theta_1 = \Lambda(K, 1)$, $\Theta_2 \sim N(0, 1)$, $\Theta_3 \sim N(0, 1)$ and $\Theta_4 \sim N(0, 1)$. Let $\kappa$ be a function that generates a new tree containing $\Delta$ splits generated using $\tilde{\Delta}$ and with $\Upsilon \sim N(0, 1)$. Then an initial forest, $\Xi^0$, is formed that contains $\bar{\Psi}$ trees each generated using $\kappa$ and with $\Gamma_0 = \text{logit}(E_{t \in \tau(y_t)})$ and $\Gamma_1 = 0.05$.

In addition to the initial forest, two scalar values, $\Phi_1$ and $\Phi_2$, are created and both set to have the value 0.01. These scalars will be updated and used to determine how large the modifications to an existing forest need to be to yield an appropriate acceptance rate. Generally, an acceptance rate of approximately 40% is thought to ensure that convergence towards the underlying distribution requires as few MPMC iterations as possible.\(^5\)

### 4.9.3 Updating

For the BEAST to update it is necessary to define a stochastic method for generating a new candidate forest from an existing forest. Here two functions, $\alpha_1$ and $\alpha_2$, are defined to perform this task. The function $\alpha_1$ takes as inputs a forest, $\Xi^\Omega$, and a scalar, $\Phi$, and outputs a new forest, $\Xi^\Psi$. An application of $\alpha_1$ is defined as follows:

---

\(^4\)A standard normal distribution is a normal distribution with mean zero and unit variance.

\(^5\)Convergence rates would likely suffer for very large acceptance rates because these would imply smaller differences between existing and new forests, such that many more accepted updates would be required to reach convergence.
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

- Update $\Gamma_0$ according to $\Gamma_0^{\Psi} = \Gamma_0^o + N(0, \Phi)$
- Update $\Gamma_1$ according to $log(\Gamma_1^{\Psi}) = log(\Gamma_1^o) + N(0, \Phi)$

The function $\alpha_2$ takes as input a forest, $\Xi^o$ and outputs a new forest, $\Xi^{\Psi}$. An application of $\alpha_2$ is defined as follows:

1. Initially set $\Xi^{\Psi}$ to be equal to $\Xi^o$.

2. Set $\gamma$ to be a vector of length $K$ for which the $k$'th element is equal to the total number of times that $\Theta_1 = k$ across all splits in $\Xi^o$.

3. Select two trees at random from those in $\Xi^{\Psi}$ and for each:
   
   (a) Select a number of splits to update, $\theta$, from $\{0, 1, \ldots, \Delta\}$, where each number has an equal probability of being drawn. Randomly select $\theta$ splits from the tree and regenerate each according to $\bar{\Delta}$ (as defined in the previous Subsection) except with $\Theta_1 = \Lambda(K, \gamma)$.

   (b) Redraw the value of $\Upsilon$ from $N(0, 1)$.

The functions $\alpha_1$ and $\alpha_2$ have been kept separate from one another so that $\Phi$ can be adjusted in order to give an appropriate acceptance rate. In this form $\alpha_2$ represents the modification of a forest’s trees while $\alpha_1$ represents the modification of the parameters in the sigmoid used to map the outputs from trees onto a probability estimate.

Once the BEAST has been initialized (previous Subsection) in order to provide a forest, $\Xi^o$, and two scalars, $\Phi_1$ and $\Phi_2$, the following steps are applied:

1. Initialize $\bar{\Xi}$ as an empty set of forests.

2. Set $\delta_1 = 0$ and $\delta_2 = 0$.

3. For $\sigma \in \{1, 2, \ldots, \Theta\}$:
4.9 The Bayesian Ensemble of Additive Sigmoidal Trees (BEAST)

(a) For $\beta \in \{1, 2\}$:

i. Generate a new candidate forest, $\Xi^a$, by applying $\alpha_\beta$ to $\Xi^o$ and $\Phi$.

ii. Calculate the likelihood ratio, $\tau = \bar{L}(\Xi^a(x_\tau), y_\tau)/\bar{L}(\Xi^o(x_\tau), y_\tau)$, where $\bar{L}$ is the likelihood function.

iii. If $\tau > 1$ set $\tau = 1$.

iv. With probability $\tau$:
   A. Set $\Xi^o = \Xi^a$.
   B. Set $\delta_\beta = \delta_\beta + 1$.

v. If $\sigma/1,000$ is an integer and $\sigma < \bar{\Theta}/2$:
   A. If $\delta_\beta > 400$ set $\Phi_\beta$ equal to $\Phi_\beta \times 1.1$, otherwise set $\Phi_\beta$ equal to $\Phi_\beta / 1.1$.
   B. Set $\delta_\beta = 0$.

(b) If $\sigma \in \{\bar{\Theta}/2, \bar{\Theta}/2 + \bar{\Gamma}, \bar{\Theta}/2 + 2\bar{\Gamma}, \ldots, \bar{\Theta}\}$ add $\Xi^o$ as a new member of $\bar{\Xi}$.

4.9.4 Making Predictions

The final set of estimates output by the BEAST are formed by averaging over the predictions output by each forest in $\bar{\Xi}$. Examples of risk estimates output by the BEAST are presented in Figure 4.11. Both plots exhibit the desired behavior of their associated learners tending to output greater risk estimates for older patients with smaller GCSs. Both plots are also associated with the desirable property that very small changes in feature values cannot result in large changes in estimated risks and look like they may be accepted by clinicians.
Fig. 4.11 Plots for the predicted risk output by the BEAST as a function of age and GCS. The color at a given location represents the predicted risk output by the learner for a patient with the associated age and GCS. The left and right plots are for learners trained on 100 and 10,000 patients, respectively.
Chapter 5

Inference

Chapter 3 described how to convert an ICU Database into a set of patient profiles and a response vector. Chapters 3 and 4 provided tools for converting a set of patient profiles and a response vector into an MPM. The present Chapter provides the remaining methods required to output an appropriate point error estimate and an associated standard error estimate to accompany the output MPM. This then allows for the definition of complete MPMCs capable of converting an ICU Database into one of the preferred output groups listed in Section 1.1.

5.1 Mortality Prediction Model Inducers (MPMIs)

An MPM can be specified as a combination of a feature extractor (Section 3.4), a preprocessor (Section 3.6), and a learner (Chapter 4). A Mortality Prediction Model Inducer (MPMI) is taken to be a function that maps a set of patient profiles and a response vector onto an MPM, and that can be specified as a unique combination of a feature extractor inducer, a preprocessor inducer, and learner inducer. To distinguish a single MPMI from a set of one or more MPMIs the former is denoted as an \( MPMI^0 \). Diagrammatic representations of MPMs and \( MPMI^0 \)s are provided in Figures 5.1 and 5.2, respectively.
The notation $\alpha_{\beta}$ is used to denote an $MPMI^0$ composed of Feature Extractor Inducer $\beta$, the Standard Preprocessor Inducer and Learner Inducer $\alpha$. For example, $RLR^2_{6}$ would consist of the Physiological (sixth) Feature Extractor Inducer, the Standard Preprocessor Inducer and Regularized Logistic Regression with a regularization constant of $\lambda = 20$. All of the $MPMI^0$s considered in this thesis are presented in Table 5.1. Notably all $MPMI^0$s that incorporate the Null Model are identical as they are independent of the design matrix and therefore do not need to specify a feature extractor inducer. Meanwhile the APACH learner provides only a single $MPMI^0$ because it can only be applied to a design matrix output by applying the Standard Feature Extractor.

An $MPMI^1$ is taken to be a set of one or more $MPMI^0$s that differ only in their associated hyper-parameter values. An $MPMI^1$ that contains more than one $MPMI^0$ is required where
5.1 Mortality Prediction Model Inducers (MPMIs)

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>NULL</th>
<th>NULL</th>
<th>NULL</th>
<th>NULL</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
<td>NN$^{1}$</td>
</tr>
<tr>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
<td>NN$^{3}$</td>
</tr>
<tr>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
<td>NN$^{5}$</td>
</tr>
<tr>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
<td>NN$^{7}$</td>
</tr>
<tr>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
<td>NN$^{9}$</td>
</tr>
<tr>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
<td>NN$^{11}$</td>
</tr>
<tr>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
<td>NN$^{13}$</td>
</tr>
<tr>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
<td>NN$^{15}$</td>
</tr>
<tr>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
<td>GMM$^{1}$</td>
</tr>
<tr>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
<td>RLR$^{0}$</td>
</tr>
<tr>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
<td>RLR$^{2}$</td>
</tr>
<tr>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
<td>RLR$^{4}$</td>
</tr>
<tr>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
<td>RLR$^{10}$</td>
</tr>
<tr>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
<td>RLR$^{20}$</td>
</tr>
<tr>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
<td>RLR$^{40}$</td>
</tr>
<tr>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
<td>RLR$^{60}$</td>
</tr>
<tr>
<td>APACH</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
<td>CART$^{1}$</td>
</tr>
<tr>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
<td>RF$^{1}$</td>
</tr>
<tr>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
<td>BART$^{1}$</td>
</tr>
<tr>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
<td>BEAST$^{1}$</td>
</tr>
</tbody>
</table>

Table 5.1 Table containing all of the $MPMI^{0}$s considered in this thesis. The column in which an $MPMI^{0}$ is placed is determined by the Feature Extractor Inducer that it incorporates. The first through sixth columns correspond to the Standard Extractor, Extended Extractor, Reduced Extractor, Basic Extractor, Demographic Extractor and Physiological Extractor, respectively, as defined in Section 3.4. The row in which an $MPMI^{0}$ is placed is determined by the learner that it incorporates, corresponding to the rows in Table 4.1.
one intends to perform hyper-parameter tuning\textsuperscript{1}. Hyper-parameter tuning is generally applied where:

\begin{itemize}
  \item No obvious default set of hyper-parameter values exists.
  \item Different hyper-parameter values can substantively affect the performance of the output MPMs.
\end{itemize}

\textsuperscript{1}Hyper-parameter tuning involves first partitioning a set of patient profiles into a training set and a testing set. Each \textit{MPMI}\textsubscript{0} from the \textit{MPMI}\textsubscript{1} is then applied to the training set in turn. The output MPMs are then used to generate predictions on the testing set. The performance of each \textit{MPMI}\textsubscript{0} is then evaluated as the distance between its vector of predictions and the response vector of the patients in the testing set. The \textit{MPMI}\textsubscript{0} which demonstrated the best performance is then selected (along with its associated hyper-parameter value).
<table>
<thead>
<tr>
<th><strong>APACH</strong></th>
<th><strong>NN</strong>&lt;sub&gt;1&lt;/sub&gt;</th>
<th><strong>NN</strong>&lt;sub&gt;2&lt;/sub&gt;</th>
<th><strong>NN</strong>&lt;sub&gt;3&lt;/sub&gt;</th>
<th><strong>NN</strong>&lt;sub&gt;4&lt;/sub&gt;</th>
<th><strong>NN</strong>&lt;sub&gt;5&lt;/sub&gt;</th>
<th><strong>NN</strong>&lt;sub&gt;6&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GMM&lt;sub&gt;1&lt;/sub&gt;</td>
<td>GMM&lt;sub&gt;2&lt;/sub&gt;</td>
<td>GMM&lt;sub&gt;3&lt;/sub&gt;</td>
<td>GMM&lt;sub&gt;4&lt;/sub&gt;</td>
<td>GMM&lt;sub&gt;5&lt;/sub&gt;</td>
<td>GMM&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
<tr>
<td>RLR&lt;sub&gt;1&lt;/sub&gt;</td>
<td>RLR&lt;sub&gt;2&lt;/sub&gt;</td>
<td>RLR&lt;sub&gt;3&lt;/sub&gt;</td>
<td>RLR&lt;sub&gt;4&lt;/sub&gt;</td>
<td>RLR&lt;sub&gt;5&lt;/sub&gt;</td>
<td>RLR&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
<tr>
<td>CART&lt;sub&gt;1&lt;/sub&gt;</td>
<td>CART&lt;sub&gt;2&lt;/sub&gt;</td>
<td>CART&lt;sub&gt;3&lt;/sub&gt;</td>
<td>CART&lt;sub&gt;4&lt;/sub&gt;</td>
<td>CART&lt;sub&gt;5&lt;/sub&gt;</td>
<td>CART&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
<tr>
<td>RF&lt;sub&gt;1&lt;/sub&gt;</td>
<td>RF&lt;sub&gt;2&lt;/sub&gt;</td>
<td>RF&lt;sub&gt;3&lt;/sub&gt;</td>
<td>RF&lt;sub&gt;4&lt;/sub&gt;</td>
<td>RF&lt;sub&gt;5&lt;/sub&gt;</td>
<td>RF&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
<tr>
<td>BEAST&lt;sub&gt;1&lt;/sub&gt;</td>
<td>BEAST&lt;sub&gt;2&lt;/sub&gt;</td>
<td>BEAST&lt;sub&gt;3&lt;/sub&gt;</td>
<td>BEAST&lt;sub&gt;4&lt;/sub&gt;</td>
<td>BEAST&lt;sub&gt;5&lt;/sub&gt;</td>
<td>BEAST&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2 Table containing all $MPMI^1$s considered in this thesis. Each $MPMI^1$ is taken outside of the third and sixth rows contain only a single $MPMI^0$ of the same form as the $MPMI^1$, except with the braces removed. For example $\{RF_1\}$ is an $MPMI^1$ that contains only $RF_1$. The $MPMI^1$s in the third and sixth rows each contain multiple $MPMI^0$s. $MPMI^1$s of the form $\{NN\}$ are shorthand for $\{NN^1, NN^3, NN^9, NN^{27}, NN^{81}, NN^{243}, NN^{729}\}$. $MPMI^1$s of the form $\{RLR\}$ are shorthand for $\{RLR^5, RLR^{10}, RLR^{20}, RLR^{40}, RLR^{80}, RLR^{160}\}$.

- The machine learning technique is quick to train, such that searching across multiple hyper-parameter values does not become computationally intractable.

All of the $MPMI^1$s considered in this thesis are presented in Table 5.2. For brevity $MPMI^0$s of the form $\{NN^1, NN^3, NN^9, NN^{27}, NN^{81}, NN^{243}, NN^{729}\}$ are henceforth denoted $\{NN\}$. Similarly, $MPMI^0$s of the form $\{RLR^5, RLR^{10}, RLR^{20}, RLR^{40}, RLR^{80}, RLR^{160}\}$ are denoted $\{RLR\}$.

An $MPMI^2$ is taken to be the complete set of all $MPMI^1$s that an MPMC applies. When an MPMC is applied to an ICU Database it will generally output an MPM, a performance point estimate and a standard error estimate for each $MPMI^1$ within its $MPMI^2$. An example of an $MPMI^2$ is $\{\{RF_1\}, \{NN_1\}\}$, which contains both the $\{RF_1\}$ and $\{NN_1\} MPMI^1$s. The introduction of $MPMI^2$s is necessary for PMCs that perform $MPMI^1$ comparisons (as is the case for those that provide the final three preferred output groups listed in Section 1.1). The nesting of $MPMI^0$s within $MPMI^1$s within $MPMI^2$s forces the selection of a single $MPMI^0$ for each $MPMI^1$. The alternative of combining all $MPMI^0$s into a single set and
then choosing that with the best performance would result in over-fitting, with \( MPMI^1 \)'s that contain a large number of \( MPMI^0 \)'s unduly favored.

## 5.2 Partitioning Matrix Inducers

A ‘partitioning matrix’ is introduced as a convenient object with which to specify how patients should be split between training and test sets in order to avoid over-fitting when evaluating the performance of \( MPMI^0 \)'s and \( MPMI^1 \)'s. A ‘partitioning matrix inducer’ takes as input a set of patient memberships\(^2\) and outputs a partitioning matrix. A ‘partitioning matrix’, \( z \), is taken to be an \( I \times A \) matrix of natural numbers where \( I \) is equal to the number of patients that are to be partitioned and \( A \) is the total number of partitions and depends on the type of partitioning matrix inducer used to generate \( z \).

Once a partitioning matrix has been generated, its influence on an MPMC can be outlined as follows: Let \( F \) be the number of unique values in \( z \) for which \( z 
eq 0 \). Then for every possible combination of \( a \in \{1, 2, \ldots, A\} \) and \( f \in \{1, 2, \ldots, F\} \), each \( MPMI^0 \) is trained using all patients for which \( z_a \neq f \) and evaluated on all patients for which \( z_a = f \). The performance of each \( MPMI^0 \) under consideration is then output as some average over its performances across the different values of \( a \) and \( f \). Complete details of how a partitioning matrix influences an MPMC are provided in Section 5.8. A summary of the partitioning matrix inducers considered in this thesis is provided in Table 5.3. Full details for these partition matrix inducers’ properties are given in Subsections 5.2.1-5.2.4.

### 5.2.1 Holdout

An application of ‘\( \alpha \% \) Patient Holdout’ to a set of \( I \) patients yields a partitioning matrix, \( z \), with \( I \) rows and \( A = 1 \) column. A total of \( \alpha \% \) of \( z \)'s elements are randomly set to one with

\(^2\)A vector for which the \( i \)'th element contains the identity of the ICU in which the \( i \)'th patient was treated.
5.2 Partitioning Matrix Inducers

<table>
<thead>
<tr>
<th>Partitioning Matrix Inducer</th>
<th>Total Partitions (A)</th>
<th>Testing Sets Per Partition (F)</th>
<th>Split On</th>
<th>% Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>40% Patient Holdout</td>
<td>1</td>
<td>1</td>
<td>Patients</td>
<td>60</td>
</tr>
<tr>
<td>4-fold Patient CV</td>
<td>1</td>
<td>4</td>
<td>Patients</td>
<td>75</td>
</tr>
<tr>
<td>10-rep 2-fold Patient CV</td>
<td>10</td>
<td>2</td>
<td>Patients</td>
<td>50</td>
</tr>
<tr>
<td>40% ICU Holdout</td>
<td>1</td>
<td>1</td>
<td>ICUs</td>
<td>60</td>
</tr>
<tr>
<td>4-fold ICU CV</td>
<td>1</td>
<td>4</td>
<td>ICUs</td>
<td>75</td>
</tr>
<tr>
<td>10-rep 2-fold ICU CV</td>
<td>10</td>
<td>2</td>
<td>ICUs</td>
<td>50</td>
</tr>
<tr>
<td>2-fold ICU CV</td>
<td>1</td>
<td>2</td>
<td>ICUs</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 5.3 Table summarizing the seven partitioning matrix inducers considered in this thesis. The second column states how many times the patients are partitioned. The third column states how many training sets are contained within a given partition. The ‘split on’ column taking the value ‘ICUs’ indicates that patients treated in the same ICU are all deliberately keep together within the same test set. The final column states the percentage of patients that make up any training set.

the remainder set to zero. Such a partition implies that each patient is assigned to either the single test set or the single training set.

Holdout is highly intuitive and allows for the construction of heuristic models with minimal issues related to over-fitting. Selection of an appropriate value of \( \alpha \) represents a trade-off, with smaller values tending to increase the performance and stability of the output MPM but also reducing the accuracy of the output MPM’s associated performance point estimate. The Standard MPM uses a compromise of 40% Patient Holdout. The primary issue with Holdout relative to the alternative partitioning matrix inducers considered in this thesis is that the limited number of patients in either or both the training and testing sets tends to limit the performance of output MPMs and/or the accuracy of performance point estimates.

5.2.2 Cross-Validation

An application of ‘\( F \)-fold patient cross-validation’ to a set of \( I \) patients yields a partitioning matrix, \( z \), with \( I \) rows and \( A = 1 \) column. The entries in the partitioning matrix are randomly assigned values from \( \{1, 2, \ldots, F\} \), with the constraint that the number of entries with each
assigned value are as equal as possible. Such a partition implies that each patient belongs to a single test set and $F - 1$ training sets.

Cross-validation allows for all patients to be used in both training and test sets so as to maximize the accuracy of the output MPMs and their associated performance point estimates. However, the performance point estimate output using cross-validation is an average over multiple internally trained MPMs rather than that measured directly using the output MPM, making its interpretation less straightforward than is the case for Holdout.

Selection of an appropriate value of $F$ represents a trade-off, with smaller values tending to increase the performance and stability of the output MPM but also increasing computation times. In the extreme case of leave-one-out cross-validation the number of partitions, $F$, is set to equal the number of patients, $I$, such that one must train $I$ MPMs. However, this approach is often prohibitively slow, especially when the $MPM^0$s are complex or $I$ is large. The other extreme is 2-fold cross-validation that requires the training of only two MPMs. However, the MPMs output by 2-fold cross-validation will tend to be considerably less accurate as each is trained on only 50% of patients. The compromise used for the majority of this thesis is 4-fold cross-validation.

### 5.2.3 Repeat Cross-Validation

Cross-validation is a stochastic process (provided $F \neq I$) with the exact MPMs trained and performance point estimates output dependent on how patients are split. To average out this variability, one can repeat the cross-validation process multiple times. An application of ‘A-repeat $F$-fold patient cross-validation’ to a set of $I$ patients yields a partitioning matrix, $z$, with $I$ rows and $A$ columns. For each column in the partitioning matrix, entries are randomly assigned values $z_i \in \{1, 2, \ldots, F\}$ under the constraint that the number of entries with each assigned value are as equal as possible within each column.
5.2 Partitioning Matrix Inducers

The main drawback associated with repeat cross-validation relative to regular cross-validation is a further increase in computation time. In this thesis 10-repeat 2-fold cross-validation is considered as a potential alternative to 4-fold cross-validation, with the two methods expected to require similar computation times.\(^3\)

### 5.2.4 Splitting on Patients vs ICUs

Partitioning matrix inducers that split on patients without considering ICU memberships will tend to allow the patients from a single ICU to be split between both training and test sets. The performance point estimates output using such partitioning matrix inducers will be appropriate for new patients assigned to the same ICUs as those used to train the MPM, but not necessarily for new patients from new ICUs. For benchmarking in particular this will tend to invalidate the assumption that MPMs cannot account for ICU quality of care and will result in overly optimistic generalization error estimates.

The alternative of splitting on ICUs as opposed to on patients is introduced as a potential solution to this issue. For each of the partitioning matrix inducers introduced in Subsections 5.2.1-5.2.3, an alternate form is provided in which ICUs are split as if they were patients, after which the entries in the partitioning matrix are assigned to be equal to the index of the partition to which the ICU of which the patient is a member was assigned.

For example, suppose one is interested in applying 2-fold ICU cross-validation to the following membership vector: \((1,1,1,2,2,2,3,3,4,4)\). This membership vector states that one has a total of ten patients from four different ICUs, and the output partitioning matrix will contain ten rows and one column. First, the four ICUs are randomly partitioned into two groups as, say, \((1,2,2,1)\); indicating that the first and fourth ICU are in the first testing set and the other two are in the second testing set. The only column of the partitioning matrix

\[^{3}\text{Assuming a machine learning technique has cubic computational complexity and that applying a machine learning technique to all patients takes a constant amount of time, } \alpha, \text{ then the time taken for 4-fold cross-validation will be } 4 \times \alpha \times (3/4)^3 = 1.7\alpha \text{ while the time taken for 10-repeat 2-fold cross-validation will be } 10 \times 2 \times \alpha \times (1/2)^3 = 2.5\alpha.\]
then becomes \((1, 1, 1, 2, 2, 2, 1, 1)\) such that all patients have been placed in the same test set as the ICUs of which they are members.

## 5.3 The Risk Array Inducer

Here the ‘Risk Array Inducer’ is introduced as a function that both trains and applies MPMs in order to generate mortality predictions, which can then be used to evaluate the performance of MPMs and ICUs. The Risk Array Inducer takes as inputs:

- A set of \(I\) patient profiles.
- A length \(I\) vector of responses, \(y\).
- An \(I \times A\) partitioning matrix, \(z\).
- An \(MPMI^2, \pi\), composed of \(M MPMI^1\)s.

The Risk Array Inducer outputs an \(I \times A \times M\) array of probabilities, \(\hat{p}\), referred to as a ‘risk array’. If the \(m\)'th \(MPMI^1\) in \(\pi\) contains only a single \(MPMI^0\), then element \(\hat{p}_{iam}\) of the risk array stores the prediction output by the application to the \(i\)'th patient profile of the MPM that was itself output by the application of the only \(MPMI^0\) in the \(m\)'th \(MPMI^1\) in \(\pi\), to the patients for whom \(z_{ia} \neq z_{ia}\). If the \(m\)'th \(MPMI^1\) contains multiple \(MPMI^0\)s, then the predictions in the risk array are taken to be those corresponding to the best performing \(MPMI^0\) in the \(m\)'th \(MPMI^1\), with the best performing \(MPMI^0\) chosen by applying the Hyper-Parameter Selector (detailed in Section 5.7). The Risk Array Inducer is applied as follows:

1. All values in \(\hat{p}\) are initially set to be missing.

2. For each combination of \(a \in \{1, 2, \ldots, A\}, f \in \{1, 2, \ldots, max(z)\}\) and \(m \in \{1, 2, \ldots, M\}\):
(a) If the \( m \)'th \( MPMI^1 \) in \( \pi \) contains only a single \( MPMI^0 \), set \( \theta \) to be the MPM output by the application of that \( MPMI^0 \) to the patients for whom \( z_a \neq f \). Otherwise, set \( \theta \) to be the MPM returned by the application of the Hyper-Parameter Selector to the patients for whom \( z_a \neq f \) and with the \( m \)'th \( MPMI^1 \) in \( \pi \) as its input \( MPMI^1 \).

(b) For all \( i \) for which \( z_{ia} = f \); the value of \( \hat{p}_{iiam} \) is set to be the risk estimate output by applying \( \theta \) to the \( i \)'th patient profile.

3. The risk array, \( \mathbf{\hat{p}} \), is then output.

Such a risk array is then in the appropriate form for the evaluation of MPM performance (as described below) and ICU performance (as described in Section 7.1). A diagrammatic representation of the Risk Array Inducer is provided in Figure 5.3.

5.4 Mortality Prediction Model Evaluators

An ‘MPM evaluator’ is a function that takes as inputs a response vector, \( y \), a membership vector, \( h \), and a vector of risk estimates, \( \mathbf{\hat{p}} \), and that outputs a scalar that summarizes the performance of the MPM that generated the risk estimates referred to as a ‘performance point estimate’. Only the two MPM evaluators required for the investigations of Chapter 6 are included in the present Section, with details of further MPM evaluators delayed until Chapter 7.

5.4.1 The Standardized Log-Likelihood

The Standardized Log-Likelihood (SLL) is an MPM evaluator that returns the negative of the log of the probability of the response given the predicted risks, divided by the number of
Fig. 5.3 Diagrammatic representation of the Risk Array Inducer. For each $MPMI^1$ in the input $MPMI^2$, the portion of the diagram within the dotted rectangle is repeated for multiple training/testing divisions, as specified by the input partitioning matrix. For a given repeat, if the $MPMI^1$ under consideration contains only a single $MPMI^0$ then the Hyper-Parameter Selector is not necessary and the required MPM is instead created through the application of the only $MPMI^0$ in the $MPMI^1$ under consideration to the training patient responses and profiles.

patients over which it was evaluated:

$$-\frac{1}{I} \sum_{i} [y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)].$$  \hspace{1cm} (5.1)$$

The SLL is a highly intuitive measure of MPM performance that is independent of the input membership vector. Smaller (more negative) values of the SLL indicate superior performance. Notably the SLL will return an infinite value if for any patient the estimated risk is 0 and the response is 1, or vice versa.
5.4.2 The $\chi^2$ Statistic

The $\chi^2$ Statistic ($\chi^2$) is an MPM evaluator that measures the difference between the expected and actual number of deaths within each ICU. Let $a_k$ be the set of patients belonging to the $k$’th ICU. Further let the observed and expected number of deaths in the $k$’th ICU be $\beta_k = \sum_{i \in a_k} y_i$ and $\gamma_k = \sum_{i \in a_k} \hat{y}_i$, respectively. Then the $\chi^2$ Statistic is given by:

$$\sum_k \frac{(\beta_k - \gamma_k)^2}{\gamma_k}.$$  (5.2)

If the estimated risk values were equal to the actual patient risks, then one would expect the $\chi^2$ Statistic to be drawn from a $\chi^2$-distribution with $K$ degrees of freedom (where $K$ is the total number of ICUs). Smaller $\chi^2$ values indicate superior performance and, unlike the SLL, the $\chi^2$ does depend on the input membership vector.

5.5 Performance Point Estimate Inducers

A performance point estimate is a measure of how well an MPM is expected to perform when applied to new patients. A ‘performance point estimate inducer’ takes as inputs:

- An $I \times A \times M$ risk array, $\hat{p}$.
- A length $I$ vector of responses, $y$.
- An MPM evaluator, $\alpha$.
- A length $I$ membership vector, $h$.

A performance point estimate inducer outputs a vector of performance point estimates, $\beta$, of length $M$, for which the $m$’th element provides the performance point estimate corresponding to the $m$’th $MPMI^1$ associated with the risk array generation, averaged over all partitions and
training-testing divisions. The two performance point estimate inducers considered in this thesis are defined in terms of how they generate their elements, \( \beta_m \), for \( m \in \{1, 2, \ldots, M\} \) as:

1. **Mean First**; For the \( i \)’th patient let \( \theta_{im} = E_\alpha(\hat{p}_{iam}) \), where each expectation is only evaluated over non-missing elements in \( \hat{p} \). Set \( \beta_m = \alpha(\theta_{m}, y) \), where \( \alpha \) applies the given MPM evaluator to the non-missing entries in \( \theta_{m} \).

2. **Mean Second** Set \( \beta_m = E_\alpha(\alpha(\hat{p}_{am}, y)) \), where \( \alpha \) applies the given MPM evaluator to the non-missing values of \( \hat{p}_{am} \).

Notably if the input risk array has \( A = 1 \) then the Mean First and Mean Second performance point estimate inducers provide the same output.

### 5.6 Standard Error Estimate Inducers

A standard error estimate is a measure of the uncertainty associated with a performance point estimate. A ‘standard error estimate inducer’ takes as inputs:

- An \( I \times A \times M \) risk array, \( \hat{p} \).
- A length \( I \) vector of responses, \( y \).
- An MPM evaluator, \( \alpha \).
- A length \( I \) membership vector, \( h \).
- An \( I \times A \) partitioning matrix, \( z \).

A standard error estimate inducer outputs a vector of standard error estimates, \( \beta \), of length \( M \), for which the \( m \)’th element provides the standard error estimate corresponding to the \( m \)’th \( MPMI^1 \) associated with the risk array generation. The three standard error estimate inducers considered in this thesis are ‘Set Comparison’, ‘Patient Bootstrapping’ and ‘ICU Bootstrapping’.
Set Comparison

Applying the Risk Array Inducer to a partitioning matrix for which \( A > 1 \) or \( E > 1 \) results in the training of multiple MPMs for each \( MPMI^0 \). In such a scenario, one can output a standard error estimate as a measure of the variability in the performance of the different MPMs within each \( MPMI^0 \). Here ‘set comparison’ is introduced as a standard error estimate inducer that embodies this approach and that is applied as follows:

1. Initialize \( \eta \) as an \( F \times A \times M \) array of zeros where \( F = \max(z) \).

2. For \( m \in \{1, 2, \ldots, M\} \), \( f \in \{1, 2, \ldots, F\} \) and \( a \in \{1, 2, \ldots, A\} \):
   
   (a) Let \( \pi \) be the set of patients for which \( z_a = f \).

   (b) Set \( \eta_{fam} = \alpha(\hat{p}_{\pi am}, y_{\pi}) \).

3. Initialize \( \beta \) as a length \( M \) vector of zeros.

4. For \( m \in \{1, 2, \ldots, M\} \) set \( \beta_m \) equal to the standard deviation of \( \eta_{.,m} \).

5. For 4-fold cross-validation set \( \beta = \beta / 1.84 \).

6. Return \( \beta \).

Step 5 is to adjust for the greater variability associated with the smaller test sets in 4-fold cross-validation. The factor of 1.84 is equal to twice the expectation of the standard deviation of four values drawn from an \( N(0, 1) \) distribution.

Set comparison has the advantage that it can account for sources of variability in the performance point estimate that arise both from the instability in the trained MPM relative to its training set and from the patients which are included in the test set. However, set comparison can only be applied if either \( A > 1 \) or \( F > 1 \) and therefore cannot be implemented with Holdout.
5.6 Standard Error Estimate Inducers

**Patient Bootstrapping**

If one makes the assumption that the contribution of the finite training set to the uncertainty in a performance point estimate is small compared to that from the finite test set, then a standard error estimate can be output as a measure of the variability in performance point estimates generated on bootstrap re-samples of the patients in the test sets. Here ‘patient bootstrapping’ is introduced as a standard error estimate inducer that implements this approach by splitting on patients as follows:

1. Initialize $\eta$ as a $20 \times M$ array of zeros.
2. Let $\theta_{im} = E_a(\hat{r}_{iam})$.
3. For $\phi \in \{1, 2, \ldots, 20\}$:
   
   (a) Set $\pi$ to be a bootstrap re-sampling of $\{1, 2, \ldots, I\}$.
   
   (b) For $m \in \{1, 2, \ldots, M\}$ set $\eta_{\phi m} = \alpha(\theta_{\pi m}, y_{\pi})$.
4. Initialize $\beta$ as a length $M$ vector of zeros.
5. For $m \in \{1, 2, \ldots, M\}$ set $\beta_m$ equal to the standard deviation of $\eta_{., m}$.
6. Return $\beta$.

**ICU Bootstrapping**

Here ‘ICU bootstrapping’ is introduced as a standard error estimate inducer that implements splitting on ICUs as follows:

1. Initialize $\eta$ as a $20 \times M$ array of zeros.
2. Let $\theta_{im} = E_a(\hat{r}_{iam})$.
3. For $\phi \in \{1, 2, \ldots, 20\}$:
(a) Set $\nu$ to be a bootstrap re-sampling of $\{1, 2, \ldots, K\}$.

(b) Set $\pi$ to be an empty vector.

(c) For $i \in \{1, 2, \ldots, I\}$ add $i$ to $\pi$ a total number of times equal to the number of times that the ICU to which the $i$'th patient belongs appears in $\nu$.

(d) For $m \in \{1, 2, \ldots, M\}$ set $\eta_{m} = \alpha(\theta_{m}, y_{\pi})$.

4. Initialize $\beta$ as a length $M$ vector of zeros.

5. For $m \in \{1, 2, \ldots, M\}$ set $\beta_{m}$ equal to the standard deviation of the $\eta_{m}$.

6. Return $\beta$.

### 5.7 Hyper-Parameter Selection

An $\text{MPMI}^1$ consists of either a single $\text{MPMI}^0$, or a set of $\text{MPMI}^0$s that each apply the same machine learning technique but with different hyper-parameter values. For the latter case, a single $\text{MPMI}^0$ must be chosen from the $\text{MPMI}^0$s within the $\text{MPMI}^1$ and this is acheived through the application of the ‘Hyper-Parameter Selector’. The ‘Hyper-Parameter Selector’ is a function that takes as inputs:

1. An $\text{MPMI}^1$, $\pi$, composed of $\Theta \text{MPMI}^0$s.

2. A set of patient profiles

3. A response vector

4. A membership vector

5. A partitioning matrix inducer

6. An MPM evaluator.
7. A performance point estimate inducer.

The Hyper-Parameter Selector outputs an $MPM$ through the application of the following steps:

1. Apply the partitioning matrix inducer to the membership vector in order to generate a partitioning matrix.

2. Form an $MPMI^2$ consisting of $\Theta$ $MPMI^1$s where for $\theta \in \{1, 2, \ldots, \Theta\}$, the $\theta$’th $MPMI^1$ contains only the $\theta$’th $MPMI^0$ from $\pi$.

3. Apply the risk array inducer to the input patient profiles, response vector, partitioning matrix and $MPMI^2$.

4. Apply the performance point estimate inducer and MPM evaluator to the risk array in order to generate a vector of $\pi$ performance point estimates.

5. Let $\sigma$ be the index corresponding to the greatest performance point estimate.

6. Apply the $\sigma$’th $MPMI^0$ from the input $MPMI^1$ to the input patient profiles and response vector and output the MPM that is generated.

A diagrammatic representation of the Risk Array Inducer is provided in Figure 5.4.

The Hyper-Parameter Selector is quite flexible, with its inputs containing a mixture of data (the patient profiles, response vector and membership vector) and options (the partitioning matrix inducer, MPM evaluator and performance point estimate inducer) in addition to the input $MPMI^1$. To simplify matters, the options passed to the Hyper-Parameter Selector are fixed such that the partitioning matrix inducer is always taken to be 4-fold patient cross-validation and the MPM evaluator is always the Standardized Log-Likelihood, while the performance point estimate inducer becomes mute because both Mean First and Mean Second are equivalent for 4-fold patient cross-validation. The two alternatives to fixing these options
5.7 Hyper-Parameter Selection

are to examine a wide range of different sets of options or to fix the options to match those that are used in an MPMC’s specification\(^4\). The former is dismissed because it represents a great deal of work that will likely yield little insight, while the latter would be problematic because some machine learning techniques require hyper-parameter selection and others do not. As such, a poor hyper-parameter selection approach would penalize only a subset of \(\text{MPMI}^1\)’s and would act as a confounder to the investigations of this thesis. The selection of

\(^4\)All MPMCs must specify a partitioning matrix inducer, MPM evaluator and performance point estimate inducer in order to evaluate the performance of the final output MPMs, and that are used entirely outside of hyper-parameter selection. The latter option then represents setting the hyper-parameter selection options to be equal to those used in the evaluation of these final output MPMs.
4-fold patient cross-validation and the Standardized Log-Likelihood is intended as a natural set of options that should perform well in most situations\(^5\).

## 5.8 MPMC Specification and Application

Now that the vast majority of MPMC steps have been introduced, it is necessary to specify how these steps fit together to create a complete MPMC. Such a specification is provided in Figure 5.5 for an MPMC that creates every type of MPMC output. In practice an MPMC will only be required to generate a subset of these four output types, with a list of useful output combinations provided at the end of Section 1.1.

The remainder of this thesis includes the application of various different MPMCs. This then motivates the inclusion of a convenient way of specifying MPMCs. Table 5.4 provides a complete list of all of the options that may be selected in order to specify an MPMC, outside of \(MPMI^2\) selection. Each MPMC in this thesis can be specified as the union of a list of options from this table and an \(MPMI^2\); for example the Standard MPMC can be denoted:

\[
(1.0, 2.1, 3.0, 4.0, 5.2, 6.1, 7.0, 9.1) \{\{APACH\}\}. \tag{5.3}
\]

However, for convenience MPMCs are provided with complete option names throughout this thesis, with the Standard MPMC presented in Table 5.5. Although it would have been straightforward to have included the \(MPMI^2\) specification using the same framework, such an approach would have required all of the \(MPMI^2\)s used in this thesis to be specified in the present Chapter - without an appreciation of the role that such \(MPMI^2\) are intended to perform.

\(^5\)The Standardized Log-Likelihood can be viewed as the performance evaluator used to select parameters when applying logistic regression. The use of the Standardized Log-Likelihood in \(\Theta\) might then be seen as placing parameter and hyper-parameter selection on an equal footing.
<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Section</th>
<th>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Response</td>
<td>3.3</td>
<td>1.0</td>
<td>Mortality at Hospital Discharge</td>
</tr>
<tr>
<td>2</td>
<td>Prediction Time Inducer</td>
<td>3.2.1</td>
<td>2.1</td>
<td>Deterministic Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.2</td>
<td>24h Fixed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3</td>
<td>Stochastic</td>
</tr>
<tr>
<td>3</td>
<td>Treatment Event Retention</td>
<td>3.2.3</td>
<td>3.0</td>
<td>Treatment Events Retained</td>
</tr>
<tr>
<td>4</td>
<td>Transfer Event Obfuscation</td>
<td>3.2.2</td>
<td>4.0</td>
<td>Transfer Events Obfuscated</td>
</tr>
<tr>
<td>5</td>
<td>Unconditional Stay Exclusions</td>
<td>3.5.1</td>
<td>5.1</td>
<td>Basic Exclusions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.2</td>
<td>Standard Exclusions</td>
</tr>
<tr>
<td>6</td>
<td>Conditional Stay Exclusions</td>
<td>3.5.2</td>
<td>6.1</td>
<td>Retain First Stay Only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.2</td>
<td>Retain One Random Stay</td>
</tr>
<tr>
<td>7</td>
<td>Preprocessor Inducer</td>
<td>3.6</td>
<td>7.0</td>
<td>Standard</td>
</tr>
<tr>
<td>8</td>
<td>Internal Partitioning Matrix Inducer</td>
<td>5.2</td>
<td>8.0</td>
<td>4-Fold Patient Holdout</td>
</tr>
<tr>
<td>9</td>
<td>External Partitioning Matrix Inducer</td>
<td>5.2</td>
<td>9.1</td>
<td>60%-Patient Holdout</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.2</td>
<td>4-Fold Patient CV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.3</td>
<td>10-Repeat 2-Fold Patient CV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.4</td>
<td>60%-ICU Holdout</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.5</td>
<td>4-Fold ICU CV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.6</td>
<td>10-Repeat 2-Fold ICU CV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9.7</td>
<td>2-Fold ICU CV</td>
</tr>
<tr>
<td>10</td>
<td>Internal MPM Evaluator</td>
<td>5.4</td>
<td>10.0</td>
<td>Standardized Log-Likelihood</td>
</tr>
<tr>
<td>11</td>
<td>Internal Performance Point Estimate Inducer</td>
<td>5.5</td>
<td>11.0</td>
<td>NA</td>
</tr>
<tr>
<td>12</td>
<td>External Performance Point Estimate Inducer</td>
<td>5.5</td>
<td>12.1</td>
<td>Mean First</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12.2</td>
<td>Mean Second</td>
</tr>
<tr>
<td>13</td>
<td>Standard Error Estimate Inducer</td>
<td>5.6</td>
<td>13.1</td>
<td>Set Comparison</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13.2</td>
<td>Individual Bootstrapping</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13.3</td>
<td>ICU Bootstrapping</td>
</tr>
<tr>
<td>14</td>
<td>ICU Performance Estimate Inducer</td>
<td>7.1</td>
<td>14.1</td>
<td>Standardized Mortality Ratio</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14.2</td>
<td>Posterior Quality Mean</td>
</tr>
</tbody>
</table>

Table 5.4 Summary of the MPMC options that are considered in this thesis. Each ‘selection’ references a different MPMC step where multiple different methods are considered feasible. The third column states the Section of this thesis in which the details of the stated selection and its options are provided. The MPMCs used in this thesis can be specified as a list of the options from this table in addition to an \( MPMI^2 \).
<table>
<thead>
<tr>
<th>Selection</th>
<th>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>1.0</td>
<td>Mortality at Hospital Discharge</td>
</tr>
<tr>
<td>Prediction Time Inducer</td>
<td>2.1</td>
<td>Deterministic Variable</td>
</tr>
<tr>
<td>Treatment Event Retention</td>
<td>3.0</td>
<td>Treatment Events Retained</td>
</tr>
<tr>
<td>Transfer Event Obfuscation</td>
<td>4.0</td>
<td>Transfer Events Obfuscated</td>
</tr>
<tr>
<td>Unconditional Stay Exclusions</td>
<td>5.2</td>
<td>Standard Exclusions</td>
</tr>
<tr>
<td>Conditional Stay Exclusions</td>
<td>6.1</td>
<td>First Stay Only</td>
</tr>
<tr>
<td>Preprocessor Inducer</td>
<td>7.0</td>
<td>Standard</td>
</tr>
<tr>
<td>External Partitioning Matrix Inducer</td>
<td>9.1</td>
<td>60%-Patient Holdout</td>
</tr>
<tr>
<td>MPMI²</td>
<td>-</td>
<td>{{APACH}}</td>
</tr>
</tbody>
</table>

Table 5.5 Specification for the Standard MPMC.

Selections associated with only a single option have been included for two reasons: firstly, they highlight places where alternative options could be included if one were interested in performing further MPMC investigations. Secondly, MPMCs may omit certain selections entirely (as is the case for the Standard MPMC with regard to selections 8 and 10-14) with the inclusion or exclusion of such options helping to specify whether or not the MPMC should apply the associated step.
Data Extraction and Stay Exclusions

ICU Database

Response Vector

Patient Profiles

Membership Vector

Partitioning Matrix

Inducer

MPMEvaluator

Performance Point Estimator

Standard Error

Estimate Inducer

MPMI

Hyper-Parameter Selector

MPM

Final MPMs

ICU Evaluator

M Quality of Care Estimator-ICU

M Standard Error Estimates

Estimate Inducer

MPMI

Fig. 5.5 Diagrammatic representation of an MPMC. The portion of the diagram within the dotted rectangle is repeated for each of the $M_{MPMI1}$. Dashed and un-dashed connections are interchangeable.
Chapter 6

Simulations

This Chapter provides a set of parametric equations able to simulate ICU Databases containing both numerous ICUs and known quality of care (QoC) values. Such simulations are then used in Chapters 7 and 8 to support investigations into different types of performance evaluators and different partitioning approaches, respectively.

Necessity of Simulations in Investigating Performance Evaluators

A good ICU evaluator aims to output ICU QoC estimates that are highly correlated with the actual ICU QoC values. If the actual QoC of ICUs is known then the performance of different ICU evaluators can be assessed by measuring the correlation between the predicted and actual ICU QoCs. However, ICU Databases do not contain direct measures of ICU QoC because such measures would require patients to be assigned to ICUs in an entirely random manner in order to average out differences in ICU case-mix. To fully investigate ICU evaluators, one must therefore simulate ICU Databases in which actual ICU QoC values are known.
Necessity of Simulations in Investigating Partitioning Approaches

A good partitioning approach should output an MPM performance point estimate that is as close as possible to the generalization error of the output MPM. If one had a database containing a very large number of ICUs then ICUs could be split into numerous blocks, each still containing a substantial number of ICUs. An MPMC applied to one such block would then output an MPM and an estimate for its generalization error. This output MPM could then be applied to the patients in the remaining blocks in order to assess its actual generalization error. The similarity between the estimated and actual generalization errors could then be used to assess the performance of a given partitioning approach. However, real-world ICU Databases contain only a limited number of ICUs. As such, simulations are required in order to inflate the number of ICUs so that the generalization error associated with MPMs can be directly measured and compared to the performance point estimates output by different MPMCs, allowing the performance of different partitioning approaches to be assessed.

Key MPM Performance Measures

Throughout this Chapter the scaled-log likelihood (SLL) and Pearson’s $\chi^2$ test statistic [64], as defined in Subsections 5.4.1 and 5.4.2 respectively, are used to measure MPM performance. To simplify discussion, it is assumed that the SLL is a measure of risking performance while the $\chi^2$ is a measure of benchmarking performance. This assumption is based on the discussion of Subsection 1.3.3 and is further supported in Chapter 7, but is not necessary for the selection of simulation parameters performed in this Chapter. Instead the simulation parameter selection relies on the assumption that the extent to which the SLL and $\chi^2$ measure each of benchmarking and risking is linearly separable.
6.1 The Simulator

The Simulator is introduced as a function that generates patients and ICUs in distinct ‘simulations blocks’, each containing 109 ICUs and 231,019 patients that have identical ICU memberships to the patients in the APACHE Database. The number of simulation blocks, $B$, can be set to be greater than unity so that the generalization error of MPMs can be measured directly. The Simulator takes as inputs:

- A set of 11 simulation parameters $\{k_1, k_2, \ldots, k_{11}\}$.
- A length $I$ membership vector, $h$.
- A partitioning matrix inducer.
- A positive integer $M$ specifying the number of $MPMI^1$s.
- A positive integer $B$ specifying the number of simulation blocks.

and outputs:
6.1 The Simulator

1. An $I \times B$ response matrix, $y$, for which $b_{yi}$ gives the response for the $i$'th patient in the $b$'th block.

2. A $K \times B$ ICU quality of care matrix, $q$, for which $b_{qk}$ gives the quality of care provided by the $k$'th ICU in the $b$'th block.

3. An $I \times B$ array of actual patient risks, $p$, for which $b_{pi}$ gives the actual probability of the response for the $i$'th patient from the $b$'th block.

4. An $I \times A \times M \times B \times B$ risk array, $\hat{p}$, for which $b_{\beta \hat{p}_{ma}}$ gives:
   
   (a) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th $MPMI^1$ evaluated on the $a$'th partition if $b = \beta$.

   (b) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th final $MPMI^1$ output on the $\beta$'th block if $b \neq \beta$.

The proceeding three Subsections provide details of how the Simulator functions as well as the motivation behind its chosen form.

### 6.1.1 Patients and ICUs

Consider the idealized situation in which an infinite number of patients and an infinite number of ICUs exist. Suppose that each patient is stochastically assigned to an ICU, with the probability of assignment to a given ICU a function of the patient’s state, and with an infinite number of patients initially assigned to each ICU.

Let the $i$'th patient’s ‘state’, $\alpha_i$, specify everything about the patient’s current physiology, and let $\kappa_k$ be the set of indices corresponding to the patients initially assigned to the $k$'th ICU. Further suppose that immediately prior to ICU admission the investigator may reassign patients at his or her discretion. All patients are then admitted to, and treated in, their assigned ICU until either death or hospital discharge, with hospital discharge representing the patient
having recovered to the extent that they no longer require ICU care, and both death and discharge assumed to be beyond the investigator’s control.

Let \( p_{ik} \) be the probability that a patient in state \( \alpha_i \) will develop the response (death before discharge) if they are treated in the \( k \)'th ICU. Let a patient’s ‘logit risk’, \( r_{ik} \), be defined as the logit of \( p_{ik} \):

\[
    r_{ik} \equiv \log \left( \frac{p_{ik}}{1 - p_{ik}} \right),
\]

where it is assumed that the \( p_{ik} \in (0, 1) \), such that the \( r_{ik} \in \mathbb{R} \).

Let \(||k_k||\) and \( I \) represent the number of patients initially assigned to the \( k \)'th ICU and the total number of patients, respectively. Then a patient’s risk is comprised of three components:

- **A Null Component**, \( n = \frac{1}{I} \sum_k ||k_k|| r_{ik} \); the risk for an average patient treated in an average ICU.

- **The Quality Component**, \( q_k = n - \frac{1}{I} \sum_i r_{ik} \); the change in risk due to treating an average patient in the \( k \)'th ICU rather than treating them in an average ICU.

- **The Patient Component**, \( \rho_i = \frac{1}{I} \sum_k ||k_k|| r_{ik} - n \); the difference in risk for the \( i \)'th patient relative to an average patient, each treated in an average ICU.

Let \( h_i \) denote the index of the ICU to which the \( i \)'th patient was originally assigned. Then the patient component can be split into two parts:

- **The Case-mix Component**, \( c_k = \frac{1}{||k_k||} \sum_{i \in k_k} \rho_i \); the extent to which the \( k \)'th ICU’s case-mix differs from that of an average ICU.

- **The Specific Component**, \( l_i = \rho_i - c_{h_i} \); the extent to which the \( i \)'th patient’s risk differs from that of an average patient from the ICU to which they were originally assigned.

It is assumed from here on that patients are always treated in the ICUs to which they are initially assigned, such that a more concise notation of the form \( p_i \equiv p_{ih_i} \) and \( r_i \equiv r_{ih_i} \) can be
used. It is further assumed that a patient’s logit risk can be expressed as a linear sum of the above components as:

\[ r_i = n - q_{hi} + \rho_i \]  \hspace{1cm} (6.2)

or equivalently:

\[ r_i = n - q_{hi} + c_{hi} + l_i \]  \hspace{1cm} (6.3)

where again the response for the \( i \)'th patient, \( y_i \), is assumed to take the value 1 (death before hospital discharge) with probability \( p_i \) and 0 (survival to hospital discharge) otherwise:

\[ Pr(y_i = 1) \equiv p_i \equiv \frac{1}{1 + e^{-r_i}} \]  \hspace{1cm} (6.4)

In reality patients exist in a variety of states, with a patient’s state influencing the ICU to which they are admitted. However explicitly modeling patient states, and the process by which patients with different states are assigning to different ICUs, is not necessary given how the simulations of this thesis are used. Instead, each ICU is generated with a different quality component and case-mix component, while patients are generated with different specific components and then randomly assigned to ICUs.

For simplicity, the ICU Quality Component, \( q_k \), Case-mix Component, \( c_k \), and Specific Component, \( l_i \), are taken to be normally distributed with a mean of 0 and variances of \( k_1^2 \), \( k_2^2 \) and \( k_3^2 \) respectively:

\[ q_k \sim N(0, k_1^2) \]  \hspace{1cm} (6.5)

\[ c_k \sim N(0, k_2^2) \]  \hspace{1cm} (6.6)

\[ l_i \sim N(0, k_3^2) \]  \hspace{1cm} (6.7)

Equations 6.3 through 6.7 then specify how a patient’s probability of death before hospital discharge is modeled in these simulations. Notably, several strong assumptions have been made in generating these equations. The linear form of Equation 6.3 does not allow for
interactions between a patient’s Specific Component and an ICU’s Quality Component, such that ICUs are modeled as performing well or poorly across all patient subgroups regardless of diagnosis or severity. Further, taking $q_k$ and $c_k$ to be drawn from independent normal distributions precludes the study of interactions between ICU case-mix and quality of care. However, these assumptions allow for a simple model in which patients can be entirely specified by their Specific Component, $l_i$, (that instantiates how ill the patient is) and their membership, $h_i$, (that states the ICU to which they are assigned), while each ICU can be entirely specified by its Case-mix Component, $c_k$, and Quality Component, $q_k$. Further, the potential interactions between the different components of a patient’s risk are not necessary for the investigations of this thesis.

Full details of a patient’s state are never known. Instead one has access to a patient profile, containing everything of clinical relevance that is known about the patient immediately prior to admission, excluding information relating to the ICU to which the patient is to be assigned. This exclusion is equivalent to ensuring that all mortality predictions generated using a patient profile are made under the assumption that the patient is about to be randomly reassigned to another ICU. This exclusion is necessary to ensure that risk predictions made using patient profiles do not account for ICU quality. If this assumption was completely invalid then the risk estimates output by MPMs could not be used for ICU benchmarking.

For simulations, the $i$’th patient’s profile is set to contain both the patient’s Specific Component, $l_i$, and the Case-mix Component of the ICU to which the patient is assigned, $c_{h_i}$. This is then an example of a perfect patient profile, in that it contains all information relevant to a patient’s risk, excluding the quality of care provided by the ICU to which the patient is to be assigned. A simpler patient profile could be defined by recombining the Specific Component and Case-mix Component back into the Patient Component. However, this is avoided because it would not allow simulated MPMs to account for each of these components to differing extents.
6.1 The Simulator

6.1.2 MPMIs

Once patients and ICUs have been independently generated for each block, the next step is to apply the same $MPMI^2$ across all blocks. For simplicity, the $M MPMI^1$s in the $MPMI^2$ applied by the Simulator are taken to each contain only a single $MPMI^0$ that applies CART as its learner inducer and with no preprocessor inducer; as such these $MPMI^0$s differ only in terms of their feature extractor inducer. Because the two-value patient profiles described in the previous Subsection differ from the patient profiles described in Section 3.2 to such an extent, the feature extractor inducers defined in Section 3.4 cannot be applied to the simulated patient profiles. Instead the ‘Simulated Feature Extractor’ is introduced which, when $M = 1$, takes as input a set of $I$ simulated patient profiles and that outputs an $I \times 2$ design matrix with first and second columns $\hat{c}$ and $\hat{l}$, respectively, calculated as:

\[
\hat{c}_i \sim N(Cc_{bh_i}, k_2^2(C(1 - C)))
\]  

(6.8)

and

\[
\hat{l}_i \sim N(Ll_i, k_2^2(L(1 - L))),
\]

(6.9)

where $k_2$ is a simulation parameter as defined in the previous Subsection and $C \in [0, 1]$ and $L \in [0, 1]$ are referred to as the ‘Case-mix Ability’ and ‘Specific Ability’ of the feature extractor and represent the extent to which it can account for the Case-mix Component and Specific Component, respectively. A Case-mix Ability of 0 indicates that the feature extractor cannot account for any case-mix, outputting $\hat{c}_i = 0$ for all $i$. A Case-mix Ability of 1 indicates that the feature extractor can perfectly account for case-mix, outputting $\hat{c}_i = c_{bh_i}$ for all $i$. A Case-mix Ability of between 0 and 1 indicates that the feature extractor will partially account for case-mix, outputting $\hat{c}_i$ as an attenuated version of $c_{bh_i}$ with added Gaussian noise.

In practice, most $MPMI^1$s are quite similar to one another in terms of the machine learning techniques and feature extractor inducers that they apply. This similarity then results
6.1 The Simulator

in predictions that are not independent, even after conditioning on the patient profiles. To mimic this behavior, it is necessary for the Simulated Feature Extractor to generate the design matrices of each \( MPMI^1 \) in unison when \( M > 1 \), such that the design matrices can be assigned values that are correlated even after conditioning on the patient profiles. To achieve this, when \( M > 1 \) the Simulated Feature Extractor is defined to be a function that takes as input a set of \( I \) simulated patient profiles and that outputs two \( I \times M \) matrices, \( \hat{c} \) and \( \hat{l} \), calculated as:

\[
\hat{c}_{im} = C_m c_{ih} + \Delta_{hk}^m k_2 \sqrt{C_m(1-C_m)},
\]

(6.10)

where

\[
\begin{pmatrix}
\Delta_1^k \\
\Delta_2^k \\
\Delta_3^k \\
\vdots \\
\Delta_M^k
\end{pmatrix} \sim MVN
\begin{pmatrix}
1 & \cdots & k^2_{10} \\
1 & & \\
\vdots & 1 & \\
k^2_{10} & \cdots & \ddots
\end{pmatrix}
\]

and

\[
\hat{l}_{im} = L_m l_i + \Lambda_{il}^m k_3 \sqrt{L_m(1-L_m)},
\]

(6.11)

where

\[
\begin{pmatrix}
\Lambda_1^i \\
\Lambda_2^i \\
\Lambda_3^i \\
\vdots \\
\Lambda_M^i
\end{pmatrix} \sim MVN
\begin{pmatrix}
1 & \cdots & k^2_{11} \\
1 & & \\
\vdots & 1 & \\
k^2_{11} & \cdots & \ddots
\end{pmatrix}
\]

and where \( C \) and \( L \) are both length \( M \) vectors for which the \( m \)'th element gives the Case-mix Ability and Specific Ability, respectively, of the \( m \)'th \( MPMI^1 \) in the Simulated \( MPMI^2 \).

The ‘Simulated Feature Extractor Inducer’ is then introduced as the feature extractor inducer used by each of the \( MPMI^0 \)'s in the \( MPMI^2 \) used to generate simulations. The
Simulated Feature Extractor Inducer is defined to be a function that takes as inputs a set of simulation parameters, \( \{k_2, k_5, k_6, k_7, k_8, k_9\} \), and that outputs the Simulated Feature Extractor for which the inverse-logit of the \( C_m \) and \( L_m \) are taken to be drawn from a bivariate normal distribution:

\[
\text{logit}^{-1}\left( \frac{C_m}{L_m} \right) \sim N\left( \begin{pmatrix} k_5 \\ k_6 \\ k_7k_8k_9 \\ k_7k_8k_9 \end{pmatrix}, \begin{pmatrix} k_2^2 & k_7k_8k_9 \\ k_7k_8k_9 & k_8^2 \end{pmatrix} \right).
\] (6.12)

The inclusion of the inverse-logit term ensures that all \( C_m \) and \( L_m \) are bounded between 0 and 1. The use of a bivariate normal distribution with a non-zero value for \( k_9 \) allows for the Case-mix Ability and Specific Ability of \( MPMI^1 \)'s to be correlated.

After the application of the Simulated Feature Extractor, the design matrix for the \( m \)'th \( MPMI^1 \) exists as an \( I \times 2 \) matrix with first and second columns provided by \( \hat{c}_m \) and \( \hat{l}_m \), respectively. When an \( MPMI^2 \) is applied to a non-simulated ICU Database, the variability in the output MPMs is split between the effects of the different learner inducers and the different feature extractor inducers that they incorporate. In the simulations of this thesis, the variability in MPMs is entirely due to the differences in the feature extraction step, with all \( MPMI^1 \)'s incorporating the same learner inducer. This choice was made to reduce complexity and was viable because the simulations were not required to compare the performance of different machine learning techniques.

When used on non-simulated ICU Databases, a learner will usually be applied to a design matrix with a sizable number (\( \sim 100 \)) of columns, while for simulations the design matrix to which the learner is applied has only two columns. For a given machine learning technique, this reduction in features will generally result in a substantial reduction in the variability of the MPM performance estimates output on different partitions. To offset this, all \( MPMI^0 \)'s applied during simulations are chosen to incorporate CART, with the expectation that this will greatly increase the diversity of the MPMs output over different partitions by the same \( MPMI^0 \), making the differences in partitioning approaches more pronounced.
6.2 The Representative MPMC

6.1.3 Risk Array Generation

Finally one must generate the required risk array, $\hat{p}$, as follows:

1. For $b \in \{1, 2, \ldots, B\}$:
   
   (a) A matrix of $MPMI^2$’s is created for which each element contains the Simulated $MPMI^2$.
   
   (b) The $\hat{b}_{p_{pp}}$ are populated as the output from applying the Risk Array Inducer (Section 5.3) to the patient profiles and response vector from the $b$’th block, and using the input partitioning matrix inducer and the matrix of $MPMI^2$’s.

2. For each combination of $b \in \{1, 2, \ldots, B\}$ and $m \in \{1, 2, \ldots, M\}$:
   
   (a) If the input partitioning matrix inducer is a form of Holdout an MPM is generated by applying the $m$’th $MPMI^1$ from the Simulated $MPMI^2$ to all patients in the training set of the $b$’th block.
   
   (b) If the input partitioning matrix inducer is not a form of Holdout an MPM is generated by applying the $m$’th $MPMI^1$ from the Simulated $MPMI^2$ to all patients in the $b$’th block.
   
   (c) For all combinations of $i \in \{1, 2, \ldots, I\}$ and $\beta \in \{1, 2, \ldots, B\}$ for which $\beta \neq b$:

   The values of $\hat{b}_{p_{m_{i}}}^{\beta}$ are each set equal to the risk estimate output by applying the MPM to the $i$’th patient in the $\beta$’th block.

6.2 The Representative MPMC

In this Section the ‘Representative MPMC’ is introduced as an example of an MPMC that contains a wide range of $MPMI^1$’s similar to those that might be applied in practice. The main purpose of the Representative MPMC is to aid in the fitting of simulation parameters such that simulations are made as realistic as possible.
6.2 The Representative MPMC

6.2.1 Specification

The requirements for the Representative MPMC are:

1. The $MPMI^2$ that it applies must contain multiple $MPMI^1$s.

2. The $MPMI^1$s should be as representative of the $MPMI^1$s that might be used in practice as possible.

3. It must be of a form that can be applied to the APACHE Database.

4. The partitioning approach it applies should provide predictions for all patients, to ensure that statistics evaluated on the risk array that it generates are as stable as possible.

Requirement 3 specifies the use of Standard Exclusions (Section 3.5.1) and First Stay Exclusions (Section 3.5.2). Both the Fixed and Deterministic Variable Prediction Time Exclusions (3.2.1) can be applied to the APACHE Database, with Deterministic Variable more appropriate for benchmarking. The main use of the Representative MPMC is in fitting simulation parameters, and these simulations are mainly used to investigate different performance evaluators and partitioning approaches, with the primary concerns relating to their use in benchmarking. As such, Deterministic Variable prediction time exclusions are selected for use in the Representative MPMC, while 2-fold ICU CV (described in Section 5.2) is chosen as the simplest partitioning matrix inducer that meets Requirement 4.

To satisfy Requirements 1 and 2, the $MPMI^2$ used in the Representative MPMC is selected to contain a wide range of $MPMI^1$s including those based on regularized logistic regression (RLR), Random Forests (RF), Bayesian Additive Regression Trees (BART), nearest neighbors (NN) and Gaussian classifiers (GC), each applied to the standard, demographic, physiological and basic features (as defined in Section 3.4). The Representative MPMC is then as defined in Table 6.1.
### 6.2 The Representative MPMC

<table>
<thead>
<tr>
<th>Selection</th>
<th>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>1.0</td>
<td>Mortality at Hospital Discharge</td>
</tr>
<tr>
<td>Prediction Time Inducer</td>
<td>2.1</td>
<td>Deterministic Variable</td>
</tr>
<tr>
<td>Treatment Event Retention</td>
<td>3.0</td>
<td>Treatment Events Retained</td>
</tr>
<tr>
<td>Transfer Event Obfuscation</td>
<td>4.0</td>
<td>Transfer Events Obfuscated</td>
</tr>
<tr>
<td>Unconditional Stay Exclusions</td>
<td>5.2</td>
<td>Standard Exclusions</td>
</tr>
<tr>
<td>Conditional Stay Exclusions</td>
<td>6.1</td>
<td>First Stay Only</td>
</tr>
<tr>
<td>Preprocessor Inducer</td>
<td>7.0</td>
<td>Standard</td>
</tr>
<tr>
<td>Internal Partitioning Matrix</td>
<td>8.0</td>
<td>4-Fold Patient Holdout</td>
</tr>
<tr>
<td>External Partitioning Matrix Inducer</td>
<td>9.7</td>
<td>2-Fold ICU CV</td>
</tr>
<tr>
<td>Internal MPM Evaluator</td>
<td>10.0</td>
<td>Standardized Log-Likelihood</td>
</tr>
</tbody>
</table>

### Table 6.1 Specification for the Representative MPMC.

Descriptions of the various steps are linked through Table 5.4.

However, this is only one example of an appropriate $MPMI^2$, with many potential alternatives available due to the ambiguous nature of Requirement 2. In the case where one is interested in the trade-off between parsimony and predictive accuracy, one might want an $MPMI^2$ that contains several feature extractor inducers that output design matrices containing very different numbers of features. By contrast, if one were only interested in accuracy, then one might want an $MPMI^2$ that contains several different state of the art machine learning techniques, but which uses only a single feature extractor outputting many features. To try and account for these different possibilities, two $MPMI^1$ subsets are defined that focus on each of these two scenarios.

The ‘Broad $MPMI^1$ Subset’ is chosen to contain all of the $MPMI^1$s in the Representative MPMC’s $MPMI^2$ excluding $\{APACH\}$ and $\{NULL\}$. This subset then contains a broad range of plausible $MPMI^1$s that might be used in the first scenario. The ‘Performance $MPMI^1$ Subset’ is chosen to contain only $\{RLR_1\}$, $\{RF_1\}$, $\{BART_1\}$ and $\{NN_1\}$. This is
commensurate with the aim of being well-suited to performance over parsimony, with each such \( MPMI^1 \) using all available features and a viable machine learning technique.

The remainder of this Section focuses on the performance of the \( MPMI^1 \)s of the Representative MPMC in terms of both their SLL and \( \chi^2 \). Insight from this focus is intended to help place limits on the values that the simulation parameters can take.

**Method of Generating the SLL and \( \chi^2 \)**

1. Initialize two length-22 vectors of zeros, \( \Psi \) and \( \Omega \).

2. Apply the Representative MPMC (defined in Table 6.1) to the APACHE Database.

3. For \( m \in \{1, 2, \ldots, 22\} \):
   
   (a) Set \( \Psi_m \) equal to the performance point estimate for the SLL associated with the \( m \)'th \( MPMI^1 \).

   (b) Set \( \Omega_m \) equal to the performance point estimate for the \( \chi^2 \) associated with the \( m \)'th \( MPMI^1 \).

4. Let \( \beta \) be a function that takes as inputs a response vector, \( y \), and a scalar, \( \alpha \in [0, 1] \), and that outputs a vector of risk estimates as:

\[
\beta(y, \alpha) = \frac{(1 - \alpha)}{I} \sum y + \alpha y.
\]  

(6.13)

5. Initialize two vectors that contain no elements, \( \bar{\Psi} \) and \( \bar{\Omega} \).

6. For \( \Theta \in \{0, 0.01, 0.02, \ldots, 1.0\} \):
   
   (a) Calculate the SLL of \( \beta(y, \Theta) \) and append the result to \( \bar{\Psi} \).

   (b) Calculate the \( \chi^2 \) of \( \beta(y, \Theta) \) and append the result to \( \bar{\Omega} \).
6.2 The Representative MPMC

Fig. 6.1 Plot of the performance of each $MPMI^1$ in the Representative MPMC evaluated on the APACHE Database. The x and y axes represent the SLL and the log of the $\chi^2$ value, respectively. The black line plots $\bar{\Psi}$ against $\bar{\Omega}$. The color and shape of the points correspond to the Feature Extractor and Learner Inducer, respectively.

6.2.2 Results and discussion

Figure 6.1 displays the performance of each $MPMI^1$ in the Representative MPMC evaluated on the APACHE Database. The x and y axes represent the SLL (that is assumed to measure risking performance) and the log of the $\chi^2$ (that is assumed to measure benchmarking performance). Points towards the bottom left of the plot correspond to better-performing $MPMI^1$s. The black line plots $\bar{\Psi}$ against $\bar{\Omega}$, with $MPMI^1$s that fall below this line better able to account for case-mix than a patient’s specific illness. The value of $\Theta$ for which the $\chi^2$ of $\beta(y, \Theta)$ is equal to that of the best performing $MPMI^1$ is at $\Theta = 0.484$. 
That the point representing the Null Model in Figure 6.1 is above and to the right of all other points indicates that all other $MPMI^1$’s considered outperform the Null Model in terms of both their ability to account for case-mix and their ability to account for patient risk. The location of APACH among the bottom left set of the points indicates that, given the information available in the APACHE Database, the APACHE IV model is an effective MPM that does a good job of accounting for both ICU case-mix and individual patient risk.

All points in Figure 6.1 are well below the black line except for the Null which must, by definition, lie on it. This indicates that MPMs tend to be considerably better at accounting for ICU case-mix than they are at accounting for a patient’s specific illness. However, it is unclear if the limited extent to which the MPMs account for patient mortality is due to them outputting poor patient risk estimates or whether the variability in patient risk is small. That the value of $\Theta$ for which the $\chi^2$ Statistic of the line is equal to that of the best performing $MPMI^1$ is at $\Theta \sim 0.5$ indicates that MPMs are able to account for approximately three quarters of the inter-ICU mortality rate variability. As MPMs cannot account for quality of care variability, this puts an upper limit of 0.25 on the proportion of inter-ICU mortality rate variability that can be due to differences in ICU quality of care. The proportion of the remaining variability that is due to each of the difference in case-mix and the difference in quality of care remains unclear. That said, $\Omega$ seems to show little sign of leveling off in Figure 6.1, suggesting that at least some proportion of the remaining variability could be accounted for by applying $MPMI^1$’s that utilize more rich patient profiles in databases that are otherwise similar to the APACHE Database. This then suggests that at least some proportion of the remaining inter-ICU mortality rate variability is due to unaccounted-for differences in case-mix. Clearly if no part of the inter-ICU mortality rate variability was due to differences in ICU quality of care, then ICU benchmarking would not be possible. As such it will be assumed that differences in both case-mix and quality of care account for
substantial proportions of the inter-ICU mortality rate variability that is left unaccounted for by the best $MPMI^1$s applied to the APACHE Database.

The strength of the association between the $\Psi$ and $\Omega$ in Figure 6.1 is only moderate. However, it is unclear to what extent this lack of correlation is due to the benchmarking and risking ability of MPMs being genuinely decoupled, versus the extent to which it is due to the inaccuracy of the stated $\Omega$ values. Each generated $\Omega$ value is effectively an average over one data-point per ICU, with only 109 ICUs in the APACHE Database. The extent to which these two effects result in this lack of correlation is considered in detail in Chapter 7.

Error bars have been omitted from Figure 6.1 because the appropriate machinery required for their generation is not introduced until Chapter 8. The lack of error bars in Figure 6.1 makes it very difficult to use this figure to make detailed comparisons of the performance of different $MPMI^1$s, although some broad trends still warrant discussion. In general, the $MPMI^1$s incorporating the Standard Extractor Inducer (red) outperform those incorporating any other Feature Extractor Inducer. $MPMI^1$s incorporating Gaussian Classifiers combined with Feature Extractors with a large number of features perform poorly (dark blue, green and red upside down triangles). This is as expected as Gaussian Classifiers are not designed for application on datasets containing large numbers of features, and supports their exclusion from the Performance $MPMI^1$ subset. $MPMI^1$s incorporating NN tend to have poor $\Psi$ values but relatively good $\Omega$, supporting their inclusion in the Performance $MPMI^1$ subset.

### 6.3 Risk Array Statistics

Here a set of nine ‘risk array statistics’ are defined, to summarize the key attributes of a risk array generated through either the application of an MPMC to an ICU Database or by the Simulator. These statistics are then evaluated on the risk arrays created by the application of the Representative MPMC to the APACHE Database for each of the Broad $MPMI^1$ Subset
and the Performance $MPMI^1$ Subset in turn, in preparation for the fitting of simulation parameter values in the next Section.

### 6.3.1 Specification

For any vector $\alpha$, let $E(\alpha)$ and $\Upsilon(\alpha)$ denote the mean and variance of $\alpha$, respectively. Let $\kappa(\beta, \gamma)$ denote the Kendall tau rank correlation [1] between any pair of equal length vectors $\beta$ and $\gamma$, with the Kendall correlation preferred here over the Pearson correlation because the former is unaffected by monotonic transformations of its input vectors. Suppose that one has a vector of responses, $y$, a membership vector, $h$, and a risk array, $\hat{y}$, of dimension $I \times M$ - generated through the application of either the Simulator with a single simulation block or an MPMC, implementing a partitioning matrix inducer incorporating only a single partition. Then the nine risk array statistics, $\{W^1, W^2, \ldots, W^9\}$, are defined as:

1. The mean survival rate:
   
   \[ W^1 = E(y). \]

2. The inter-ICU mortality rate variability:
   
   \[ W^2 = \chi^2(E(y), y, h). \]

   This is a measure of the total inter-ICU mortality rate variability due to a combination of differences in ICU case-mix and differences in ICU quality of care.

3. The mean MPM specific performance:
   
   \[ W^3 = E(\Psi), \quad \text{where} \quad \Psi_m = SLL(\hat{y}^m, y). \]

   This is a measure of the extent to which an average $MPMI^1$ can account for the specific component of an average patient’s risk.
4. The variability in specific performance of MPMs:

\[ W^4 = \Upsilon(\Psi), \quad \text{where} \quad \Psi_m = SLL(\hat{y}^m, y). \]

This is a measure of the variability in the extent to which MPMI\(^1\)s can account for the specific component of an average patient’s risk.

5. The mean MPM case-mix performance:

\[ W^5 = E(\Omega), \quad \text{where} \quad \Omega_m = \log(\chi^2(\hat{y}^m, y)). \]

This is a measure of the extent to which an average MPMI\(^1\) can account for the case-mix component of an average patient’s risk.

6. The variability in the case-mix performance of MPMs:

\[ W^6 = \Upsilon(\Omega), \quad \text{where} \quad \Omega_m = \log(\chi^2(\hat{y}^m, y)). \]

This is a measure of the variability in the extent to which MPMI\(^1\)s can account for the case-mix component of an average patient’s risk.

7. The performance correlation:

\[ W^7 = \kappa(\Psi, \Omega), \quad \text{where} \quad \Psi_m = SLL(\hat{y}^m, y) \quad \text{and} \quad \Omega_m = \log(\chi^2(\hat{y}^m, y)). \]

This is a measure of the extent to which an MPMI\(^1\) that is better than average at accounting for the specific component also tends to be better than average at accounting for inter-ICU variability.
8. The specific similarity:

\[ W^8 = E(\pi), \quad \text{where} \quad \pi_{\eta} = \frac{1}{M-1} \sum_{m \neq \eta} (\zeta_{\eta m}) \quad \text{and} \quad \zeta_{\eta m} = \kappa(\hat{y}^m, \hat{y}^\eta). \]

This is the average correlation between the specific component of predictions made by different MPMI\(^1\)s.

9. The case-mix similarity:

\[ W^9 = E(\pi), \quad \text{where} \quad \pi_{\eta} = \frac{1}{M-1} \sum_{m \neq \eta} (\eta_{\eta m}), \]

\[ \zeta_{\eta m} = \kappa(\upsilon^m, \upsilon^\eta) \quad \text{and} \quad \upsilon^\tau_k = \frac{\sum_i \bar{I}(h_i = k)\hat{y}_i^\tau}{\sum_i \bar{I}(h_i = k)}. \]

This is the average correlation between the case-mix component of predictions made by different MPMI\(^1\)s.

The final two risk array statistics, \( W^8 \) and \( W^9 \), are included so that the extent to which the predictions output by different MPMI\(^1\)s are conditionally dependent, given a patient’s profile, can be accounted for. This is important, because assuming that the predictions of MPMI\(^1\)s are conditionally independent would imply that an optimal MPMI\(^1\) could be generated simply by combining a large number of poor MPMI\(^1\)s.

### 6.3.2 Evaluation

The two sets of risk array statistics evaluated on the Broad and Performance MPMI\(^1\) Subsets of the Representative MPMC, \( W^B \) and \( W^P \), respectively, are calculated as follows:

1. Initialize \( \eta \) as a \( 10 \times 9 \times 2 \) array of zeros.

2. Set \( \alpha \) to contain the set of indices corresponding to the MPMI\(^1\)s in the Broad MPMI\(^1\) Subset.
3. Set $\beta$ to contain the set of indices corresponding to the $MPMI^1$s in the Performance $MPMI^1$ Subset.

4. For $\gamma \in \{1, 2, \ldots, 10\}$:
   
   (a) Apply the Representative MPMC (defined in Table 6.1) to the APACHE Database and retain the $I \times 22$ risk array, $\hat{y}$, that it outputs.

   (b) Set $\eta_{\gamma, 1}$ equal to the values output by evaluating the risk array statistics on $\hat{y}_{,\alpha}$.

   (c) Set $\eta_{\gamma, 2}$ equal to the values output by evaluating the risk array statistics on $\hat{y}_{,\beta}$.

5. Set $W_B$ and $W_P$ to be length-9 vectors of zeros.

6. For $\theta \in \{1, 2, \ldots, 9\}$:
   
   (a) Set $W_B^{\theta}$ equal to the median of $\eta_{,j1}$.

   (b) Set $W_P^{\theta}$ equal to the median of $\eta_{,j2}$.

7. Output the $W_B$ and $W_P$.

### 6.3.3 Results and discussion

The risk array statistics evaluated on the Representative MPMC are presented in Table 6.2. That $W_3$ and $W_5$ are smaller in the fourth column than in the third column indicates that, as expected, the Performance $MPMI^1$ Subset tends to offer better risking and benchmarking performance, respectively, than the Broad $MPMI^1$ Subset. The primary use of the values reported in Table 6.2 is in the fitting of simulation parameters performed in the next Section, while these values offer little insight beyond what has already been discussed with reference to Figure 6.1. Perhaps the only notable exception is the assessment of $W_6^p$ and $W_7^p$, for which discussion is delayed until the next Section.
### 6.4 Parameter Fitting

The focus of this Section is on assigning appropriate values to the simulation parameters, so that the risk array statistics evaluated on the outputs of the Simulator are similar to those evaluated on the outputs of the application of the Representative MPMC to the APACHE Database.

#### 6.4.1 Specification

The eleven simulation parameters together offer a total of eleven degrees of freedom. Of these eleven, a total of nine shall be constrained so that the risk array statistics evaluated on the outputs of the Simulator are similar to those evaluated on the outputs of the Representative MPMC, while the remaining two must be fixed using expert knowledge, because they cannot be inferred from the properties of the risk arrays output by the Simulator.

One of the two risk array-independent degrees of freedom can be expressed as the relative value of $k_1^2 / (k_1^2 + k_2^2)$ and $k_5$. Of these two terms, the former represents the fraction of the

<table>
<thead>
<tr>
<th>Risk Array Statistic</th>
<th>Interpretation</th>
<th>$MPMI^1$ Subset</th>
<th>$W^B$</th>
<th>$W^P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_1$</td>
<td>$E(y)$</td>
<td>0.122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W_2$</td>
<td>$E(\Omega_{Null})$</td>
<td>8.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W_3$</td>
<td>$E(\Psi)$</td>
<td>0.277</td>
<td></td>
<td>0.246</td>
</tr>
<tr>
<td>$W_4$</td>
<td>$\Upsilon(\Psi)$</td>
<td>0.0272</td>
<td></td>
<td>0.0165</td>
</tr>
<tr>
<td>$W_5$</td>
<td>$E(\Omega)$</td>
<td>7.24</td>
<td></td>
<td>6.79</td>
</tr>
<tr>
<td>$W_6$</td>
<td>$\Upsilon(\Omega)$</td>
<td>0.265</td>
<td></td>
<td>0.0484</td>
</tr>
<tr>
<td>$W_7$</td>
<td>$\kappa(\Psi,\Omega)$</td>
<td>0.284</td>
<td></td>
<td>0.167</td>
</tr>
<tr>
<td>$W_8$</td>
<td>$\kappa(\hat{p},\hat{p})$</td>
<td>0.710</td>
<td></td>
<td>0.874</td>
</tr>
<tr>
<td>$W_9$</td>
<td>$\kappa(E_{\phi},\hat{p},E_{\phi},\hat{p})$</td>
<td>0.858</td>
<td></td>
<td>0.965</td>
</tr>
</tbody>
</table>

Table 6.2 Table to display the values of the risk array statistics evaluated on the Representative MPMC. The entries in the third and fourth columns correspond to the Broad and Performance $MPMI^1$ Subsets, respectively. The symbols in the interpretation column have the following meanings: $E$ is the expectation, $\Upsilon$ is the variance, $\Omega$ is the log of the $\chi^2$ value, $\Psi$ is the SLL, $\kappa$ is the Kendall correlation, and $\phi_k$ denotes the set of patients in the $k$’th ICU.
total variance in inter-ICU mortality that is accounted for by differences in ICU quality of
care (as opposed to differences in ICU case-mix), while the latter represents the average
Case-mix Ability of MPMI\(^1\)s. Although the value of \(k_1^2 + k_2^2\) can be inferred from the second
risk array statistic, \(W^2\), a given value of \(W^5\) (the average MPM case-mix performance) can be
accounted for by various combinations of \(k_5\) and \(k_2^2/(k_1^2 + k_2^2)\) values. It could be that a large
part of the variability in unaccounted for inter-ICU mortality rates is due to differences in ICU
quality of care (QoC), and that the MPMI\(^1\)s are able to account for nearly all of the case-mix
variability. Alternatively, it could be that only a small part of the variability in unaccounted
for inter-ICU mortality rates is due to differences in ICU QoC, and that the MPMI\(^1\)s are able
to account for a smaller fraction of the more substantial case-mix variability. The issue is
that the risk array, and hence risk array statistics, are identical in both cases.

In Subsection 6.2.2 an upper limit of 0.25 was placed on \(k_1^2/(k_1^2 + k_2^2)\), with the actual
value supposed to be a considerable distance from the bounds of the valid [0 0.25] range.
Because of the importance of the value of \(k_1^2/(k_1^2 + k_2^2)\), two values for this ratio are considered,
so that the effects of changing the value of this ratio can be investigated. The values
considered for \(k_1^2/(k_1^2 + k_2^2)\) are 0.2 and 0.05. These correspond to the two scenarios set out
in the previous paragraph with regards to differences in QoC and the ability of MPMI\(^1\)s to
account for case-mix variability. For brevity the ratios \(k_1^2/(k_1^2 + k_2^2) = 0.2\) and \(k_1^2/(k_1^2 + k_2^2) =\)
0.05 are rearranged into \(k_2^2/k_1^2 = 4\) and \(k_2^2/k_1^2 = 19\), respectively.

A similar situation arises for the second of the two risk array-independent degrees of
freedom. The values of \(k_3^2\) and \(k_6^2\) specify the extent to which intra-ICU patient risk varies
and the extent to which an average MPM accounts for this variability, respectively. It could
be that the value of \(k_3^2\) is large and that \(k_6^2\) is small, such that there is a lot of variability in
patient risk but that the average MPM does a poor job of accounting for it. Alternatively
it could be that the value of \(k_3^2\) is small and that \(k_6^2\) is large. However, unlike with \(k_5\) and
\(k_1^2/(k_1^2 + k_2^2)\), the relative values of \(k_3^2\) and \(k_6^2\) are of little consequence and for convenience \(k_3\)
is set to be 2.5 for the remainder of this thesis. The selected values of $k_1^2/(k_1^2 + k_2^2)$ and $k_3$
then provide two classes of simulation parameters:

- $k^4$ for which $k_3 = 2.5$ and $k_2^2/k_1^2 = 4$.
- $k^{19}$ for which $k_3 = 2.5$ and $k_2^2/k_1^2 = 19$.

Let $\Omega(k, M, 1)$ output a length nine vector by applying the risk array statistics to the
outputs of the Simulator run with simulation parameters $k$, $B = 1$ simulation block, $M$
$MPMI^1$s and using 2-fold ICU CV. As the simulations are stochastic, the value of $\Omega(k, M, 1)$
will also be stochastic. For integer $\Theta$ let $\Omega(k, M, \Theta)$ provide an output as follows:

1. Initialize $\eta$ as a $\Theta \times 9$ vector of zeros.
2. For $\theta \in \{1, 2, \ldots, \Theta\}$ set $\eta_\theta = \Omega(k, M, 1)$.
3. Set $\pi$ to be length-9 vector of zeros.
4. For $\beta \in \{1, 2, \ldots, 9\}$ set $\pi_\beta$ equal to the median of $\eta_\beta$.
5. Output $\pi$.

The purpose of $\Omega(k, M, n)$ with $n > 1$ is to act as a more stable version of $\Omega(k, M, 1)$, with
the median chosen over the mean because the risk array statistics are potentially from highly
skewed distributions.

Combining each of $k^4$ and $k^{19}$ with the Broad $MPMI^1$ Subset (defined in Section 6.2.1) then yields two sets of simulation parameter values:

- $k^{B4}$; defined to be the member of $k \in k^4$ for which $\Omega(k, 20, \infty) = W^B$
- $k^{B19}$; defined to be the member of $k \in k^{19}$ for which $\Omega(k, 20, \infty) = W^B$

where the values of $W^B$ were reported in Table 6.2. The equivalent sets of simulation parameter values, $W^P$, for the Performance $MPMI^1$ Subset (defined in Section 6.2.1) require
6.4 Parameter Fitting

a more careful definition. The Performance $MPMI^1$ Subset includes only the four best performing $MPMI^1$s, namely regularized logistic regression, Random Forests, Bayesian Additive Regression Trees and Nearest Neighbors, with each applied to the Standard Feature Set. The values of $W_3^P$ and $W_5^P$ represent the mean, and $W_4^P$, $W_6^P$ and $W_7^P$ represent the covariance, of the bivariate normal distribution defined by the risking and benchmarking performance of these four $MPMI^1$s, as embodied in the lowest four points in Figure 6.1. Clearly $W_7^P$, the correlation between these four points, is an extremely unreliable measure of the correlation of the distribution from which these points were drawn. For example, removing the $MPMI^1$ incorporating Nearest Neighbors would greatly increase the value of $W_7^P$. Similar concerns apply to $W_6^P$, with the variance derived from only four $\chi^2$ values likely to be highly inaccurate. To avoid these issues, two narrow classes of simulation parameters are defined that effectively use $W_6^B$ and $W_7^B$ in place of $W_6^P$ and $W_7^P$, as:

- $k_4^N$ for which $k_3 = 2.5$, $k_2^2/k_1^2 = 4$, $k_9 = k_9^B$ and $k_7/k_8 = k_7^B/k_8^B$

- $k_19^N$ for which $k_3 = 2.5$, $k_2^2/k_1^2 = 19$, $k_9 = k_9^B$ and $k_7/k_8 = k_7^B/k_8^B$,

where the $k_9 = k_9^B$ entry sets the correlation between the Case-mix Ability and Specific Ability for the simulations parameter set associated with the Performance $MPMI^2$ to be equal to that for simulations associated with the Broad $MPMI^2$. The $k_7/k_8 = k_7^B/k_8^B$ entry sets the ratio of variances in Case-mix and Specific Ability for simulations set using the Performance $MPMI^2$ to be equal to that for simulations set using the Broad $MPMI^2$. Combining each of $k_4^N$ and $k_19^N$ with the Performance $MPMI^1$ Subset then yields two further sets of simulation parameter values:

- $k_4^P$; defined to be the member of $k \in k_4^N$ for which $\Omega(k, 4, \infty) = W^P$

- $k_19^P$; defined to be the member of $k \in k_19^N$ for which $\Omega(k, 4, \infty) = W^P$

The approach is then to find $k_4^B$, $k_19^B$, $k_4^P$ and $k_19^P$ by stochastically searching the simulation parameter space, with a new set of simulation parameter values, $k_{\text{new}}$, preferred over an
old set of simulation parameter values, $k_{old}$, if $\Delta(\Omega(k_{new}, M, n), W) < \Delta(\Omega(k_{old}, M, n), W)$ for some measure of distance, $\Delta$.

### 6.4.2 Methods

For two arbitrary sets of risk array statistic values, $\Psi$ and $\omega$, and a set of indices, $\tau$, the distance between $\Psi$ and $\omega$ is taken to be:

$$\Delta(\Psi, \omega, \tau) = \sum_{\nu \in \tau} |\log(\Psi_{\nu}) - \log(\omega_{\nu})|$$  \hspace{1cm} (6.14)

where $\tau = \{1, 2, \ldots, 9\}$ is appropriate for the Broad $MPM^{I}$ Subset and $\tau = \{1, 2, 3, 4, 5, 8, 9\}$ is appropriate for the Performance $MPM^{I}$ Subset.

Let $\Psi$ be a function that takes as inputs a set of simulation parameters, $k$, and a binary value, $\xi \in \{0, 1\}$, and that outputs a perturbation of $k$ according to the following steps:

1. Draw a single value, $\epsilon$, from $N(0, 0.05)$.

2. Set $\log(k_1) = \log(k_1) + \epsilon$ and set $\log(k_2) = \log(k_2) - \epsilon$.

3. Set $k_4 = k_4 + N(0, 0.01)$.

4. Set $\text{logit}(k_5) = \text{logit}(k_5) + N(0, 0.05)$.

5. Set $\text{logit}(k_6) = \text{logit}(k_6) + N(0, 0.05)$.

6. Set $\text{logit}((1 + k_{10})/2) = \text{logit}((1 + k_{10})/2) + N(0, 0.05)$.

7. Set $\text{logit}((1 + k_{11})/2) = \text{logit}((1 + k_{11})/2) + N(0, 0.05)$.

8. If $\xi = 0$:

   (a) Set $\log(k_7) = \log(k_7) + N(0, 0.05)$.

   (b) Set $\log(k_8) = \log(k_8) + N(0, 0.05)$. 
(c) Set \( \logit((1+k_0)/2) = \logit((1+k_0)/2) + N(0,0.05) \).

9. If \( \xi = 1 \):
   
   (a) Draw a single value, \( \varepsilon \), from \( N(0,0.05) \).
   
   (b) Set \( \log(k_8) = \log(k_8) + \varepsilon \) and set \( \log(k_9) = \log(k_9) - \varepsilon \).

10. Return \( k \).

When applying \( \Psi \) it is necessary to set \( \xi = 0 \) when considering a Broad \( MPMI^1 \) Subset or to set \( \xi = 1 \) when considering a Performance \( MPMI^1 \) Subset.

Let \( \Upsilon(k, W, M, \tau, \xi) \) be a function that consists of the following steps:

1. Set \( k_{old} = k \).

2. Set \( \alpha = \Delta(\Omega(k_{old}, M, 100), W, \tau) \).

3. While \( \alpha > 0.1 \):
   
   (a) Set \( k_{new} = \Psi(k_{old}, \xi) \).
   
   (b) Set \( t = \Delta(\Omega(k_{new}, M, 100), W, \tau) \).
   
   (c) If \( t < \alpha \):
       
       i. Set \( k_{old} = k_{new} \).
       
       ii. Set \( \alpha = t \).

4. Output \( k_{old} \).

Then the four required sets of simulation parameter values are calculated as:

\[
\begin{align*}
    k^{B4} &= \Upsilon((0.5, 0.5, 2.5, -3.65, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5), W^B, 20, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, 0) \\
       & \quad \text{(6.15)}
\end{align*}
\]
6.4 Parameter Fitting

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Interpretation</th>
<th>Parameter Set</th>
<th>k^{B4}</th>
<th>k^{B19}</th>
<th>k^{P4}</th>
<th>k^{P19}</th>
</tr>
</thead>
<tbody>
<tr>
<td>k_1</td>
<td>Variability in Quality of Care</td>
<td></td>
<td>0.29</td>
<td>0.147</td>
<td>0.29</td>
<td>0.145</td>
</tr>
<tr>
<td>k_2</td>
<td>Variability in Case-mix</td>
<td></td>
<td>0.58</td>
<td>0.64</td>
<td>0.58</td>
<td>0.63</td>
</tr>
<tr>
<td>k_3</td>
<td>Variability in intra-ICU risk</td>
<td></td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>k_4</td>
<td>Intercept</td>
<td></td>
<td>-3.65</td>
<td>-3.65</td>
<td>-3.65</td>
<td>-3.65</td>
</tr>
<tr>
<td>k_5</td>
<td>Average Case-mix Ability</td>
<td></td>
<td>0.95</td>
<td>0.55</td>
<td>2.30</td>
<td>1.30</td>
</tr>
<tr>
<td>k_6</td>
<td>Average Random Ability</td>
<td></td>
<td>0.68</td>
<td>0.68</td>
<td>1.90</td>
<td>1.58</td>
</tr>
<tr>
<td>k_7</td>
<td>Variability in Case-mix Ability</td>
<td></td>
<td>0.90</td>
<td>0.22</td>
<td>0.85</td>
<td>0.165</td>
</tr>
<tr>
<td>k_8</td>
<td>Variability in Random Ability</td>
<td></td>
<td>0.78</td>
<td>0.78</td>
<td>0.70</td>
<td>0.59</td>
</tr>
<tr>
<td>k_9</td>
<td>Correlation between Case-mix Ability and Random Ability</td>
<td></td>
<td>0.39</td>
<td>0.41</td>
<td>0.42</td>
<td>0.41</td>
</tr>
<tr>
<td>k_{10}</td>
<td>Additional correlation between case-mix estimates</td>
<td></td>
<td>0.80</td>
<td>0.66</td>
<td>0.92</td>
<td>0.88</td>
</tr>
<tr>
<td>k_{11}</td>
<td>Additional correlation between patient specific risk estimates</td>
<td></td>
<td>0.40</td>
<td>0.36</td>
<td>0.60</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 6.3 Values taken by the simulation parameter sets. The $k^{B4}$ and $k^{B19}$ columns are for simulations fit to the Broad $MPMI^1$ Subset using the assumptions that there is either a large or small amount of ICU QoC performance variability, respectively. Similarly the $k^{P4}$ and $k^{P19}$ columns are for simulations fit to the Performance $MPMI^1$ Subset using the assumptions that there is either a large or small amount of ICU QoC performance variability, respectively.

\[
k^{B19} = \Upsilon(\frac{0.5}{\sqrt{19}}, 0.5, 2.5, -3.65, 0.5, 0.5, 0.5, 0.5, 0.5, k^{B4}_8, 20, \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, 0) \tag{6.16}
\]

\[
k^{P4} = \Upsilon((\frac{0.5}{2}, 0.5, 2.5, -3.65, 0.5, 0.5, k^{B4}_8, 0.5, k^{B19}_8, 0.5, 0.5), \Upsilon^{P4}, 4, \{1, 2, 3, 4, 5, 8, 9\}, 1) \tag{6.17}
\]

\[
k^{P19} = \Upsilon((\frac{0.5}{\sqrt{19}}, 0.5, 2.5, -3.65, 0.5, 0.5, k^{B19}_8, 0.5, k^{B19}_8, 0.5, 0.5), \Upsilon^{P4}, 4, \{1, 2, 3, 4, 5, 8, 9\}, 1) \tag{6.18}
\]

6.4.3 Results and discussion

The final values taken by the simulation parameter sets are provided in Table 6.3. Although the primary use of these parameters is in supporting the analyses of Chapters 7 and 8, significant insight can be gained from their examination.

The value of $k_9$ states the correlation between the Case-mix Ability and Random Ability of MPMs and indicates the extent to which risking performance evaluators such as the AUC can be used to measure benchmarking performance. The outputs from benchmarking
performance evaluators will tend to be much less stable than those of risking performance evaluators, especially within an MPMC that is applied to a database containing only a small number of ICUs. As such, a value of \( k_9 \) of near unity would support the use of risking performance evaluators over benchmarking performance evaluators for assessing benchmarking performance. However, the value of \( k_9 \approx 0.4 \) is well below unity and suggests that benchmarking performance evaluators are likely to outperform risking performance evaluators in terms of assessing the benchmarking performance of MPMs in most scenarios. That the value of \( k_9 \) is well below unity further motivates the need to examine the relative merits of risking and benchmarking performance evaluators as is done in Chapter 7.

The expected case-mix quality of a simulated \( MPMI^1 \) is given by \( \text{logit}^{-1}(k_5) \), and 95\% of \( MPMI^1 \)s are expected to have a case-mix quality within \( \text{logit}^{-1}(k_5 \pm 1.96k_7) \). For \( k^B_4 \), these values yield an expectation of 0.71 and an interval of [0.35 0.92], indicating that the vast majority of case-mix is accounted for by the best MPMs. For \( k^B_19 \) the equivalent expectation and interval are 0.63 and [0.39 0.82], indicating that the majority of case-mix is still accounted for by the best MPMs. The lower value of the lower bound for case-mix quality of MPMs from \( k^B_19 \) relative to \( k^B_4 \) occurs because the standard deviation in the Case-mix Quality must be approximately the same for both simulation parameter sets, while the standard deviation of \( \text{logit}^{-1}(N(b,1)) \), for arbitrary \( b \), will monotonically reduce as \( b \) moves away from 0, such that in order to maintain a larger value of \( k_5 \) a greater value of \( k_7 \) is required. This then makes extreme values such as 95\% intervals even more extreme.

One might then wonder if a measure associated with the properties of the tails of the MPM benchmarking performance distribution might be added to the risk array statistics in order to empirically assess the relative values of \( k_1 \) and \( k_2 \). However, the current set of 109 ICUs would be too small to reliably assess such tail properties, and even if one had a vast number of ICUs, assessing the relative values of \( k_1 \) and \( k_2 \) in such a way would be relying far too heavily on the simplifying assumption that the Case-mix Quality of ICUs is drawn from the
inverse logit of a normal distribution. In reality, any differences would be more likely due
to flaws in the choice of distributions used in the simulations than actual differences in the
relative values of $k_1$ and $k_2$. 
Chapter 7

Performance Evaluators

ICU evaluators are needed to assess the quality of care provided by one or more ICUs. MPM evaluators are required to assess how useful the risk estimates output by an MPM are likely to be when used in either risking or benchmarking. For the case of risking, the performance of an MPM is given by the accuracy of the risk predictions that it generates on new patients. For benchmarking, its performance is given by the extent to which its predictions can be used, in conjunction with an ICU evaluator, to accurately measure the quality of care provided by different ICUs.

In general, an ICU Database will not contain direct measures of actual ICU quality of care. Instead, these quality of care values must be estimated using a combination of risk estimates, memberships and responses. However, in simulations these risk estimates, memberships and responses are supplemented with actual quality of care values. The simulations of Chapter 6 can therefore be used to find the correlation between the estimates output by different performance evaluators and the underlying performance values that they are intended to estimate.

The structure of this Chapter is as follows: Section 7.1 investigates ICU quality of care evaluators and their use. Section 7.2 discusses the MPM evaluators used in the Standard
ICU evaluators are used to assess the risk-adjusted quality of care provided by ICUs. Such quality of care estimates can be used by ICUs internally to identify deficient areas, track progress or compare their performance against similar ICUs, or externally to help calculate appropriate levels of remuneration.\(^1\)

This Section is divided into three parts: Subsection 7.1.1 introduces the Standardized Mortality Ratio (SMR) and looks at how it can be used to infer ICU quality of care. Subsection 7.1.2 details the potential limitations of the current use of the SMR as an ICU evaluator. Subsection 7.1.3 presents an alternative ICU evaluator that is intended to address the limitations of the SMR.

### 7.1.1 The Standardized Mortality Ratio (SMR)

The Standardized Mortality Ratio (SMR) is the most common ICU evaluator used in the MPM literature \(^2\) and is defined as the ratio of the observed and expected number of deaths within a given ICU. The SMR for the \(k\)’th ICU is given by:

\[
SMR = \frac{\sum_{i \in \alpha_k} (y_i)}{\sum_{i \in \alpha_k} (\hat{y}_i)},
\]

(7.1)

where \(\alpha_k\) is the set of indices of patients belonging to the \(k\)’th ICU.

Consider a situation in which one has a large ICU and a perfect benchmarking MPM: an MPM with both a Case-mix Ability and Specific Ability of unity. In the notation of

\(^1\)As occurs in Ireland through the use of diagnosis-related groups \(^8\) in which patients are partitioned by illness according to the tenth revision of the International Statistical Classification of Diseases and Related Health Problems \(^60\)
Subsection 6.1.1 such a perfect MPM would output predictions of the form:

\[ \hat{r}_i = n + c_{hi} + l_i \] (7.2)

Combining Equation 7.2 with Equation 6.3 and assuming that \( \hat{r}_i \) and \( r_i \) are small then yields:

\[
q_{hi} = \hat{r}_i - r_i \\
= -\log(e^{r_i}/e^{\hat{r}_i}) \\
\approx -\log(r_i/\hat{r}_i) \\
\approx -\log(\logit^{-1}(r_i)/\logit^{-1}(\hat{r}_i)) \\
\approx -\log \left( \frac{\sum_{i \in \alpha} (y_i)}{\sum_{i \in \alpha} (\hat{y}_i)} \right) \\
\approx -\log(SMR).
\] (7.3)

This means that the negative log of the SMR may be interpreted as a direct measure of an ICU’s quality of care. However, this result does not hold when evaluating the SMR on an ICU with only a small number of patient profiles, because the contribution to differences in expected and actual mortality rates due to statistical noise is then no longer negligible. This is usually resolved by instead working in terms of the probability that the measured SMR would be at least as extreme were it to be evaluated on the risk estimates from a perfect benchmarking MPM. One can then infer that ICUs with an SMR that differs significantly from unity are providing either superior or inferior quality of care than an average ICU. However, as discussed in the next Subsection, this approach to ICU quality of care inference does not work when benchmarking MPMs fail to account for even a small portion of ICU case-mix.
7.1 ICU Evaluators

7.1.2 Issues with the SMR

Consider two ICUs, ‘α’ and ‘β’, where α recorded 12 deaths out of 100 patients when 10 deaths were expected and β recorded 1,200 deaths out of 10,000 patients when 1,000 deaths were expected. Both α and β will have the same SMR value of 1.2, but the (one-sided) p-value for α is 0.2 while that for β it is $4 \times 10^{-11}$. From a Bayesian viewpoint, the p-value is akin to the proportion of the posterior for which the quality of care, $q$, is greater than zero, with the differences in p-values due to the much greater evidence for a negative value of $q$ available for ICU β.

Next, consider dropping the perfect benchmarking MPM assumption so that some proportion of ICU case-mix remains unaccounted for. Then differences in SMRs for different ICUs can be due to differences in both ICU quality of care and the unaccounted for case-mix. These effects are illustrated in Figures 7.1 and 7.2. Each column of panels represents a different combination of prior and evidence. Let $q$ and $c$ be the quality of care and unaccounted for case-mix, respectively, associated with a given ICU. Then the first column of each Figure represents the situation in which one is quite certain that there is no unaccounted for case-mix ($c = 0$), while the second (third) column represents the situation in which one expects there to be a modest (substantial) amount of unaccounted for case-mix. These prior beliefs are represented as heat-maps in the top panel rows. Each column of panels in Figure 7.1 corresponds to a situation in which an ICU reported 12 deaths out of 100 patients when 10 deaths were expected. Each column of panels in Figure 7.2 corresponds to a situation in which an ICU reported 1,200 deaths out of 10,000 patients when 1,000 deaths were expected. The evidence provided by these observations is represented in the second row of panels. The posteriors arising from these situations are then presented in the third row of panels. The fourth row of panels provides the conditional distribution for the ICU quality, $q$, and can be used to evaluate the p-value of interest by calculating the total density for which $q > 0$. 
Fig. 7.1 Figure illustrating the effects of different priors on quality of care p-values in the presence of minimal evidence (an ICU that recorded 12 deaths out of 100 patients when 10 deaths were expected). The top three rows of panels provide heat-maps for the prior, evidence and posterior for the unaccounted for case-mix, $c$, and quality of care, $q$. The bottom row of panels provides the conditional distribution for $q$ and can be used to evaluate the p-value of interest by calculating the total density for which $q > 0$. The columns of panels from left to right correspond to increasing degrees of uncertainty in $c$. 
Fig. 7.2 Figure illustrating the effects of different priors on quality of care p-values in the presence of substantial evidence (an ICU that recorded 1,200 deaths out of 10,000 patients when 1,000 deaths were expected). The top three rows of panels provide heat-maps for the prior, evidence and posterior for the unaccounted for case-mix, c, and quality of care, q. The bottom row of panels provides the conditional distribution for q and can be used to evaluate the p-value of interest by calculating the total density for which q > 0. The columns of panels from left to right correspond to increasing degrees of uncertainty in c.
On moving from left to right in Figure 7.1, it can be seen that weakening the assumption that there is no unaccounted for case-mix has a relatively modest effect when one has little evidence relating to ICU performance. However, moving from left to right in Figure 7.2 illustrates that weakening this assumption has a considerable effect when one has substantial evidence relating to ICU performance. In this case, having a substantial quantity of unaccounted for case-mix, as is presented in the third column of panels, increases the p-value from a highly statistically significant $1.9 \times 10^{-10}$ to an insignificant 0.22. Given that it seems unlikely that any real-world MPM will fully account for case-mix, this result calls into question the use of ICU quality of care p-values in their current form. This then motivates the inclusion of the ‘Posterior Quality Threshold’ (defined in the Subsection below) as an ICU evaluator that does not rely so heavily on the perfect benchmarking MPM assumption.

### 7.1.3 The Posterior Model

Here the ‘Posterior Model’ is introduced as an ICU evaluator that uses a Bayesian framework to account for the residual ICU case-mix that MPMs are unable to account for. The Posterior Model assumes that the $k$’th ICU has a quality of care, $q_k$, and a degree of unaccounted for case-mix, $\mu_k$. If one knew the values of $q_k$ and $\mu_k$, then the expected mortality rate for the $k$’th ICU would be given by:

$$
\frac{1}{|\alpha_k|} \sum_{i \in \alpha_k} \logit^{-1}(\logit(\hat{y}_i) - \mu_k + q_k)
$$

(7.4)

where $\alpha_k$ denotes the set of patients belonging to the $k$’th ICU. The Posterior Model assumes that the $\mu_k$ and $q_k$ are drawn from mean zero normal distributions with variances $(1 - \sigma)\Omega^2$ and $\sigma\Omega^2$, respectively, where $\Omega$ is a positive real value and $\sigma \in [0, 1]$. The value of $\sigma$ is taken to be user-specified, with a default of 80%. For each ICU, the Posterior Model then

---

$^2$This value is set to match the ratio expected for a high performance MPM, as output by simulations (Chapter 6) using simulation parameter set $k^{P4}$ (as defined in Section 6.4).
outputs the posterior for $q_k$, with the ‘Posterior Quality Mean’ and ‘Effective p-value’ for the $k$’th ICU defined as the mean of this posterior and the proportion of this posterior with a positive sign, respectively.

For internal benchmarking, the Posterior Quality Means and Effective p-values can be used as measures, and measures of the significance, of the quality of care provided by a given ICU. A similar approach can be used for external benchmarking, although one must output both $\Omega$ and $\sigma$ in addition to the usual external benchmarking outputs listed in Section 1.1 in order to apply the Posterior Quality Mean to new ICUs.

### 7.2 Mortality Prediction Model Evaluators

An MPM evaluator is a method for assessing the performance of a mortality prediction model. Such performance estimates can be used to choose which of various MPMs is most appropriate or to determine whether a given MPM is sufficiently accurate for a given task. The identification of better MPM evaluators should lead to the identification of better MPMs, which could then help to identify exceptional ICUs and produce viable patient deterioration alarms.

The structure of this Section is as follows: Subsection 7.2.1 revisits the use of the Area Under the Curve and Hosmer-Lemshow Statistic as the MPM evaluators of the Standard MPMC. Subsection 7.2.2 details how these MPM evaluators may not be appropriate measures of an MPM’s benchmarking performance. Subsection 7.2.3 introduces alternative MPM evaluators that aim to overcome these benchmarking-related issues. Subsection 7.2.4 introduces alternative MPM evaluators that aim to focus on an MPM’s risking performance. The MPM evaluators introduced in this Section are then quantitatively compared in Section 7.3, in terms of their ability to measure each of an MPM’s risking and benchmarking performance.
7.2 Mortality Prediction Model Evaluators

7.2.1 The Standard MPM Evaluators

The Area Under the Curve (AUC)

The Area Under the Receiver Operating Characteristic Curve (AUC) is the primary MPM evaluator of the Standard MPMC. The AUC can be interpreted as the probability that the estimated risk assigned to a randomly chosen survivor is less than that assigned to a randomly chosen non-survivor. The AUC is unaffected by monotonic transformations of the estimated risk and therefore cannot measure model calibration. It can take values between 0 and 1 inclusive, with the latter indicating perfect discrimination; an AUC of 0.5 indicates that an MPM is entirely unable to discriminate.

The Hosmer-Lemshow Statistic (H-L)

The Hosmer-Lemshow (H-L) Statistic is the secondary MPM evaluator of the Standard MPMC. The H-L groups patients into deciles of expected risk and compares the expected and actual number of deaths within each decile. Let \( \eta_i \) be the set of patients whose risks are within the \( i \)’th risk decile and let the observed and expected number of deaths in the \( i \)’th decile be \( \Omega_i = \sum_{i \in \eta_i} y_i \) and \( \Psi_i = \sum_{i \in \eta_i} \hat{y}_i \), respectively. Then the H-L Statistic is given by \( \sum_i (\Omega_i - \Psi_i)^2 / \Psi_i \). If the estimated risk values were equal to the actual patient risks then one would expect the H-L value to be drawn from a \( \chi^2 \) distribution with ten degrees of freedom. The H-L Statistic is strongly affected by sample size [49] and will tend to prefer the Null Model with added jitter over all other MPMs.

Combining the AUC and H-L

When the Standard MPMC is used to compare MPMs, the usual approach is to choose the MPM with the best discrimination out of all MPMs with good calibration. This is usually achieved as follows:

1. The AUC and H-L of each MPM under consideration are evaluated.
2. The statistical significance of each H-L Statistic is calculated against the null hypothesis that all risk estimates are output by perfect MPMs.

3. The MPMs associated with an H-L Statistic that is significant at the 5% level are removed from consideration.

4. The best MPM is chosen as that which has the greatest AUC out of the retained MPMs.

One alternative to choosing the MPM with the best discrimination out of all MPMs with good calibration is to use a single MPM evaluator that considers both calibration and discrimination simultaneously. Four examples of such alternative MPM evaluators are introduced in Subsection 7.2.4, ready for the quantitative comparisons against the AUC and H-L performed in Section 7.3.

### 7.2.2 Issues with the Standard MPM Evaluators

In Subsection 1.3.3, it was suggested that the AUC might not be an appropriate performance measure for benchmarking MPMs, because of how it incorrectly selected an inferior age-based benchmarking MPM over the Null Model. The present Subsection considers additional reasons against the use of the AUC for benchmarking, that rely on the equations introduced in Chapter 6.

In Subsection 7.1.1 it was stated that an ideal benchmarking MPM should make predictions according to Equation 7.2. In risking, one is ultimately interested in the value of the patient risk, \( p_i \), such that an ideal risking MPM would output predictions of the form:

\[
\hat{r}_i = n - q_{hi} + c_{hi} + l_i
\]  

(7.5)

However, combining this equation with Equation 6.3 simply yields \( \hat{r}_i = r_i \), from which no information about an ICU’s quality of care can be inferred. As such, an ideal risking MPM cannot be used to assess ICU quality of care. This makes the ideal MPM for risking useless.
in terms of benchmarking, and provides an extreme example in which an MPM’s risking ability is a poor measure of its benchmarking ability. In the more general scenario in which MPMs are imperfect, the risking performance of the \( m \)’th MPM\(^1 \) is given by:

\[
E_i((r_i - \hat{r}_{im})^2) = E_i((c_{hi} - \hat{c}_{hi},m)^2) + E_i((l_i - \hat{l}_{im})^2) + E_i(q_i^2) = (1 - C_m)k_2^2 + (1 - L_m)k_3^2 + k_1^2,
\]

while the benchmarking performance is given by:

\[
E_k((q_k - \hat{q}_{km})^2) = E_k((E_i \in \alpha_k(\hat{c}_{hi} - c_{hi}))^2) + E_k((E_i \in \alpha_k(\hat{l}_i - l_i))^2) = (1 - C_m)k_2^2 + E_k(1/|\alpha_k|)(1 - L_m)k_3^2 \approx (1 - C_m)k_2^2 + \frac{K}{I}(1 - L_m)k_3^2,
\]

where \( \alpha_k \) is the set of indices of patients belonging to the \( k \)’th ICU. In general, one might expect the following set of equalities to hold in the vast majority of ICU setting: \( k_3^2 \gg k_2^2 \gg k_1^2K/I \). This then suggests that the contribution of the Specific Ability, \( L_m \), will tend to dominate an MPM’s risking performance, while the contribution of the Case-Mix Ability, \( C_m \), will tend to dominate an MPM’s benchmarking performance.

The primary issue is then that it is unclear to what extent the AUC and H-L focus on each of an MPM’s specific ability and case-mix ability. The expectation is that the AUC focuses on the specific ability and that it is therefore a poor measure of MPM benchmarking performance. However, this must be confirmed (Section 7.3) and alternative MPM evaluators provided (Subsections 7.2.3 and 7.2.4) before one can replace the use of the AUC in the evaluation of MPMs.
7.2 Mortality Prediction Model Evaluators

7.2.3 Membership-Dependent MPM Evaluators

The evaluation of the AUC and the H-L Statistic does not require knowledge of patient memberships. However, the inclusion of membership information might help increase the influence of the case-mix ability relative to that of the specific ability. On a patient level the Specific Component will be much larger than the Case-mix Component, such that MPM evaluators that do not consider patient memberships will tend to focus on the Specific Component. However, averaging over patients from the same ICU will leave the Case-mix Component intact (because it represents a systematic error) while the contribution of the Specific Component will reduce in proportion to the square root of the number of patients (because it represents a random error). As such, MPM evaluators that take into account patient memberships may be able to focus on the Case-mix Component and measure benchmarking performance more directly. In this Subsection, two simple membership-dependent MPM evaluators are provided.

The $\chi^2$ Statistic

Let $\alpha_k$ denote the set of patients belonging to the $k$’th ICU. Let the observed and expected number of deaths in the $k$’th ICU be $\beta_k = \sum_{i \in \alpha_k} y_i$ and $\gamma_k = \sum_{i \in \alpha_k} \hat{y}_i$, respectively. Then the $\chi^2$ Statistic is given by $\sum_k (\beta_k - \gamma_k)^2 / \gamma_k$. If the estimated risk values were equal to the actual patient risks, one would expect the H-L statistic to be drawn from a $\chi^2$ distribution with $K$ degrees of freedom, where $K$ is the total number of ICUs. The $\chi^2$ Statistic is identical to the H-L Statistic except that it splits patients by ICUs rather than by deciles of risk.

The Posterior Bias Variance

The value of $\Omega^2$ in the Posterior Model (Subsection 7.1.3) represents the total amount of inter-ICU variability that an MPM is unable to account for. Given that a good benchmarking MPM is one that can account for a large amount of ICU case-mix variability, the Posterior
Model can be used as an MPM evaluator that outputs the ‘Posterior Bias Variance’, $\Omega^2$, as a measure of an MPM’s benchmarking performance, and for which smaller values indicate superior performance.

### 7.2.4 Further Membership-Independent MPM Evaluators

The $\chi^2$ Statistic and Posterior Bias Variance were introduced in the previous Subsection as two examples of MPM evaluators that rely on patient memberships in order to assess the benchmarking performance of MPMs. In the present Subsection four additional MPM evaluators are introduced that do not rely on patient memberships, as potential alternatives to the AUC with which to measure the risking performance of MPMs.

#### The Accuracy

The ‘Accuracy’ is perhaps the simplest and most intuitive MPM evaluator. The Accuracy is evaluated as the proportion of $\hat{y}$ values that are within 0.5 of the response:

$$\frac{1}{I} \sum_{i} (|y_i - \hat{y}_i| < 0.5).$$

(7.6)

The Accuracy can be thought of as minimizing the loss associated with poor decisions in the scenario where false positives and false negatives are equally bad.

#### The Standardized Log-Likelihood

The Standardized Log-Likelihood (SLL) is another intuitive MPM evaluator and is calculated as the negative of the log of the probability of the response given the predicted risks, divided by the number of patients:

$$- \frac{1}{I} \sum_{i} [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)].$$

(7.7)
Using the SLL as an MPM evaluator with which to compare MPMs can be interpreted as selecting the MPM that predicted the actual outcome with the greatest degree of certainty. The SLL will output an infinite value if the estimated risk for any patient is 0 when the response is 1, or vice versa.

**The Brier Score**

The Brier Score outputs the root mean square error between prediction and response:

$$\frac{1}{I} \sum_i (y_i - \hat{p}_i)^2.$$  \hspace{1cm} (7.8)

The Brier Score behaves in a similar manner to the SLL, except that it has a less clear interpretation and only outputs finite values. The Brier Score might be thought of as a somewhat inappropriate porting of the root mean square error from a real response onto a binary one.

**The Boosting Loss**

The Boosting Loss is defined as:

$$\frac{2}{I} \sum_i \sqrt{\frac{|y_i - \hat{p}_i|}{1 - |y_i - \hat{p}_i|}}.$$  \hspace{1cm} (7.9)

The boosting loss can be thought of as selecting the MPM who’s predictions minimize the expected loss contingent on the following assumptions:

1. There exists some constant loss associated with a false negative.

2. There exists some constant loss associated with a false positive.

3. An uninformative prior is placed over the ratio of these two constants.
7.3 Assessing the Performance of Performance Evaluators

The Boosting Loss strongly resembles the SLL in that it has a clear interpretation, and will output an infinite value if the estimated risk for any patient is 0 when the response is 1, or vice versa.

7.3 Assessing the Performance of Performance Evaluators

This Section uses simulations to quantitatively assess the performance of both ICU and MPM performance evaluators. This is achieved by measuring the correlation between the predicted and actual performance values for each of ICU and MPM performance.

7.3.1 Specification

In this Section the performance of the MPMs is measured in terms of their ability to perform each of the five following tasks:

1. External risking
2. Internal benchmarking using the SMR
3. Internal benchmarking using the Posterior Quality Mean
4. External benchmarking using the SMR
5. External benchmarking using the Posterior Quality Mean

Throughout this Subsection it is assumed that the Simulator has been applied using 2-fold ICU CV with $B = 2$, $M = 2$ and using one of the four sets of simulation parameter values in order to generate:

1. An $I \times 2$ response matrix, $y$, for which $^b_{y_i}$ gives the response for the $i$’th patient in the $b$’th block.
2. A length $I$ membership vector, $h$.

3. A $K \times 2$ ICU quality of care matrix, $q$, for which $^b q_k$ gives the quality of care provided by the $k$'th ICU in the $b$'th block.

4. An $I \times 2$ array of actual patient risks, $p$, for which $^b p_i$ gives the actual probability of the response for the $i$'th patient from the $b$'th block.

5. An $I \times 2 \times 2 \times 2$ risk array, $\hat{p}$, for which $^b \hat{p}_i^m$ gives:
   
   (a) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th $MPMI^1$ if $b = d$.
   
   (b) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th final $MPMI^1$ output on the $d$'th block if $b \neq d$.

Further let $\kappa_{\alpha}(\beta_\alpha, \gamma_\alpha)$ be the Kendell correlation between two arbitrary vectors $\beta$ and $\gamma$ over some index $\alpha$.

### MPM Risking Performance

For risking, a good MPM is one that provides risk predictions which are strongly correlated with the actual risk. The risking performance of the $m$'th MPM trained on block $b$ and evaluated on block $\beta$ is taken to be:

$$^b \beta \alpha_m = \kappa_{\beta}(^b \hat{p}_i^m, ^b p_i)$$

with the index of the best risking MPM given by:

$$^b \beta \bar{\delta} = \begin{cases} 
1 & \text{if } ^b \beta \alpha_1 > ^b \beta \alpha_2 \\
2 & \text{otherwise} 
\end{cases}$$
ICU Benchmarking Performance

Let $\hat{q}_k^i(\hat{p}, h)$ denote the predicted quality of care for the $k$th ICU, output by applying the $i$th ICU evaluator to the $\hat{p}$ output by a single MPM for a single block. Then a good ICU evaluator should output $\hat{q}_k^i(\hat{p}, h)$ values that are highly correlated with the $q_k$ of the ICUs from the same block. As such, the performance of the $i$th ICU evaluator and $m$th MPM trained on block $b$ and evaluated on block $\beta$ is taken to be;

$$b_{\beta}^i \eta_m^1 = \kappa_k(\hat{q}_k^i(b_{\beta}^m p, h), b_{\beta} q_k).$$

(7.11)

with the index of the best benchmarking MPM given by:

$$b_{\beta}^i \delta_m^1 = \begin{cases} 1 & \text{if } b_{\beta}^i \eta_1^1 > b_{\beta}^i \eta_2^1 \\ 2 & \text{otherwise} \end{cases}$$

MPM Evaluator Outputs

If larger values for the $w$th ICU evaluator indicate superior performance then let $b_{\beta}^i \Omega_m^w$ denote the value returned by applying the $w$th MPM evaluator to the $b_{\beta}^i \hat{p}^m$, otherwise let $b_{\beta}^i \Omega_m^w$ denote the negative of this value. Further let $b_{\beta}^i \delta_m^w$ denote the index of the $MPMI^1$ that the $w$th MPM evaluator deems to be best, such that:

$$b_{\beta}^i \delta_m^w = \begin{cases} 1 & \text{if } b_{\beta}^i \Omega_1^w > b_{\beta}^i \Omega_2^w \\ 2 & \text{otherwise} \end{cases}$$

MPM Evaluator Discrimination

One potential method for measuring an MPM evaluator’s performance is to evaluate how often a given MPM evaluator fails to select the best MPM for a given task. This then yields a total of five terms, with each corresponding to one of the five tasks listed at the beginning of
7.3 Assessing the Performance of Performance Evaluators

this Section and introduced in the same order as follows:

\[ wG_1 = 1 - E_{\beta \neq b}(E_b(b_\delta^{w} = b_\delta^{1})) \]  \hspace{1cm} (7.12)

\[ wG_2 = 1 - E_b(b_\delta^{w} = b_\delta^{1}) \]  \hspace{1cm} (7.13)

\[ wG_3 = 1 - E_b(b_\delta^{w} = b_\delta^{2}) \]  \hspace{1cm} (7.14)

\[ wG_4 = 1 - E_{\beta \neq b}(E_b(b_\delta^{w} = b_\delta^{1})) \]  \hspace{1cm} (7.15)

\[ wG_5 = 1 - E_{\beta \neq b}(E_b(b_\delta^{w} = b_\delta^{2})) \]  \hspace{1cm} (7.16)

**MPM Evaluator Correlation**

Another potential method for measuring an MPM evaluator’s performance is to evaluate the average performance of the MPMs that the given evaluator selects for a given task. This then yields an additional five terms, with each corresponding to one of the five tasks listed at the beginning of this Section and introduced in the same order as follows:

\[ wZ_1 = E_{\beta \neq b}(E_b(b_\alpha^{\tilde{\theta}_w})) \]  \hspace{1cm} (7.17)

\[ wZ_2 = E_b(b_\beta^{1}) \]  \hspace{1cm} (7.18)

\[ wZ_3 = E_b(b_\beta^{2}) \]  \hspace{1cm} (7.19)

\[ wZ_4 = E_{\beta \neq b}(E_b(b_\eta^{1}_{\tilde{\theta}_w})) \]  \hspace{1cm} (7.20)

\[ wZ_5 = E_{\beta \neq b}(E_b(b_\eta^{2}_{\tilde{\theta}_w})) \]  \hspace{1cm} (7.21)

In addition to the \( w \in \{1, \ldots, 8\} \) associated with the MPM evaluators of Section 7.2, three additional MPM evaluators are included that could not or would not be used in practice. \(^9\mathbf{Z}\) values are evaluated on the predictions output by the best of the two MPMs for the task under
consideration. \( \text{values are evaluated on the predictions output by one of the two MPMs chosen at random. Finally,} \ \text{values are defined as the} \ Z \ \text{values applied to the predictions output by the Null model.} \)

### 7.3.2 Methods

The discrimination and correlation (as defined in the previous Subsection) of MPM evaluators are evaluated for various sets of simulation parameter values as follows:

1. Initialize \( G^{k^{4P}}, G^{k^{19P}} \) and \( G^{k^{4B}} \) as \( 8 \times 5 \) matrices of zeros.
2. Initialize \( Z^{k^{4P}}, Z^{k^{19P}} \) and \( Z^{k^{4B}} \) as \( 11 \times 5 \) matrices of zeros.
3. For every combination of \( k \in \{k^{4P}, k^{19P}, k^{4B}\} \) and \( \beta \in \{1, 2, \ldots, 500\} \):
   (a) Apply the Simulator with \( B = 2, M = 2, \) using 2-fold ICU CV and simulation parameter set \( k \).
   (b) Calculate \( G \) and \( Z \) as defined in the previous Subsection.
   (c) Set \( G^k = G^k + G/500 \)
   (d) Set \( Z^k = Z^k + Z/500 \)

### 7.3.3 Results and discussion

The performance of the different MPM evaluators are presented in Tables 7.1-7.5. Table 7.1 details the discrimination of MPM evaluators evaluated using simulation parameter set \( k^{4P} \). Table 7.2 combines extracts from Table 7.1 and the equivalent results evaluated using simulation parameter sets \( k^{19P} \) and \( k^{4B} \) for the most interesting MPM evaluators. Tables 7.3, 7.4 and 7.5 present the mean performance of the MPMs selected via different approaches relative to the tasks defined in Subsection 7.3.1, evaluated using simulation parameter sets \( k^{4P}, k^{19P} \) and \( k^{4B} \), respectively.
7.3 Assessing the Performance of Performance Evaluators

<table>
<thead>
<tr>
<th>MPM Evaluator</th>
<th>Risking</th>
<th>SMR</th>
<th>PQM</th>
<th>External SMR</th>
<th>PQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>2.0</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>Brier Score</td>
<td>1.2</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>HL</td>
<td>40</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>SLL</td>
<td>1.0</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>AUC</td>
<td>1.2</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>Boosting</td>
<td>0.9</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>37</td>
<td>13.9</td>
<td>14.0</td>
<td>16.9</td>
<td>15.7</td>
</tr>
<tr>
<td>PVM</td>
<td>37</td>
<td>12.1</td>
<td>12.2</td>
<td>14.5</td>
<td>13.5</td>
</tr>
</tbody>
</table>

Table 7.1: Table to display the discrimination provided by different MPM evaluators. Each entry gives the percentage of times that the given MPM evaluator fails to select the best MPM for a given task. Simulation parameter set $k^p$ was used to generate the simulations from which these results were derived. The SMR (Standardized Mortality Ratio) and PQM (Posterior Quality Mean) indicate which ICU evaluator was implemented when performing ICU benchmarking.

From Table 7.1 it is clear that the MPM evaluators considered in this thesis fall into three groups: the Accuracy, Brier Score, SLL, AUC and Boosting Loss offer excellent risking performance evaluation but poor benchmarking performance evaluation; the $\chi^2$ and PVM provide good benchmarking performance evaluation but poor risking performance evaluation; and the H-L Statistic offers very little.

The MPM evaluators in the high risking performance group are able to select the best risking MPM all but 1% of the time. Given the excellent discrimination of risking MPMs offered by every member of this group, it seems of little consequence which of the Brier Score, SLL, AUC and Boosting Loss is used, although it should be noted that the use of the AUC would fail to account for potentially important differences in model calibration if used to compare uncalibrated models. The lower performance offered by the Accuracy is likely due in part to the APACHE Database having an unbalanced response ($\sim$0.12), as well as the effective rounding off of risk information that the Accuracy performs.
7.3 Assessing the Performance of Performance Evaluators

<table>
<thead>
<tr>
<th>MPM Evaluator</th>
<th>Simulation Parameter Set</th>
<th>Risking</th>
<th>G / % Error</th>
<th>Benchmarking Internal SMR</th>
<th>Benchmarking External SMR</th>
<th>Benchmarking Internal PQM</th>
<th>Benchmarking External PQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLL</td>
<td>k^{4P}</td>
<td>1.0</td>
<td></td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>SLL</td>
<td>k^{19P}</td>
<td>0.9</td>
<td></td>
<td>40</td>
<td>40</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>SLL</td>
<td>\chi^2</td>
<td>1.6</td>
<td></td>
<td>40</td>
<td>40</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>\chi^2</td>
<td>k^{4P}</td>
<td>37</td>
<td></td>
<td>13.9</td>
<td>14.0</td>
<td>16.9</td>
<td>15.7</td>
</tr>
<tr>
<td>\chi^2</td>
<td>k^{19P}</td>
<td>36</td>
<td></td>
<td>22.8</td>
<td>22.9</td>
<td>23.9</td>
<td>25.2</td>
</tr>
<tr>
<td>\chi^2</td>
<td>k^{4B}</td>
<td>38</td>
<td></td>
<td>13.9</td>
<td>13.8</td>
<td>12.6</td>
<td>12.6</td>
</tr>
</tbody>
</table>

Table 7.2 Table to display the discrimination provided by different MPM evaluators. Each entry gives the percentage of times that the given MPM evaluator fails to select the best MPM for a given task. The simulation parameter set used to generate the simulations is given in the second column. The SMR (Standardized Mortality Ratio) and PQM (Posterior Quality Mean) indicate which ICU evaluator was implemented when performing ICU benchmarking.

The MPM evaluators in the high risking performance group are quite poor in terms of discriminating between benchmarking MPMs, with the best benchmarking MPM only selected approximately 63% of the time, while the high benchmarking performance group offers a substantial improvement with the correct selection made approximately 87% of the time. This suggests that assessing the benchmarking performance of MPMs using \chi^2 type evaluators in place of the AUC and the H-L might greatly aid the selection of superior benchmarking MPMs.

Comparing the bottom two rows of Table 7.1 indicates that the PVM offers a modest increase in benchmarking MPM discrimination relative to the \chi^2. However, the increase in complexity and the loss of familiarity that most will find on going from the \chi^2 to the PVM makes the overall benefit of the transition unclear. When used internally, the improvement associated with moving from 13.9% for the combination of the SMR and \chi^2 value to 12.2% for the combination of the PQM and PVM is relatively modest compared to the transition from 16.9% to 13.5% when used externally. The external benchmarking performance also tends to be inferior to the internal benchmarking performance, and this trend may be an
7.3 Assessing the Performance of Performance Evaluators

<table>
<thead>
<tr>
<th>MPM Selection Approach</th>
<th>Risking</th>
<th>Internal SMR</th>
<th>Internal PQM</th>
<th>External SMR</th>
<th>External PQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
<td>-</td>
<td>28.3</td>
<td>28.4</td>
<td>28.3</td>
<td>28.4</td>
</tr>
<tr>
<td>Average</td>
<td>73.6</td>
<td>50.1</td>
<td>50.6</td>
<td>50.4</td>
<td>51.0</td>
</tr>
<tr>
<td>Best</td>
<td>78.2</td>
<td>53.2</td>
<td>53.7</td>
<td>53.5</td>
<td>54.2</td>
</tr>
<tr>
<td>Accuracy</td>
<td>78.2</td>
<td>51.4</td>
<td>51.9</td>
<td>51.7</td>
<td>52.2</td>
</tr>
<tr>
<td>Brier Score</td>
<td>78.2</td>
<td>51.4</td>
<td>51.8</td>
<td>51.7</td>
<td>52.2</td>
</tr>
<tr>
<td>HL</td>
<td>75.3</td>
<td>50.4</td>
<td>50.8</td>
<td>50.8</td>
<td>51.4</td>
</tr>
<tr>
<td>SLL</td>
<td>78.2</td>
<td>51.4</td>
<td>51.8</td>
<td>51.7</td>
<td>52.2</td>
</tr>
<tr>
<td>AUC</td>
<td>78.2</td>
<td>51.4</td>
<td>51.8</td>
<td>51.7</td>
<td>52.2</td>
</tr>
<tr>
<td>Boosting</td>
<td>78.2</td>
<td>51.4</td>
<td>51.8</td>
<td>51.7</td>
<td>52.2</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>75.5</td>
<td>52.9</td>
<td>53.5</td>
<td>53.2</td>
<td>53.8</td>
</tr>
<tr>
<td>PVM</td>
<td>75.5</td>
<td>53.0</td>
<td>53.5</td>
<td>53.3</td>
<td>53.9</td>
</tr>
</tbody>
</table>

Table 7.3 Table to display the mean performance of the MPMs selected by the given selection approach relative to the tasks defined in Subsection 7.3.1. Simulation parameter set $k_{19P}$ was used to generate the simulations from which these results were derived. The SMR (Standardized Mortality Ratio) and PQM (Posterior Quality Mean) indicate which ICU evaluator was implemented when performing ICU benchmarking.

artifact caused by the use of CART: The internal estimates are made as an average over the performance of two MPMs each trained and tested on 54 or 55 ICUs, while the external estimates are made using a single MPM trained and tested on 109 ICUs. As such the former is effectively performing bagging over weak learners while the latter is not, such that the partitioning used in internal benchmarking offers superior discrimination.

Within a given simulation parameter set, the trends in Table 7.2 largely match those seen in Table 7.1. Comparing rows 1 and 3 indicates that the SLL offers worse risking MPM discrimination when simulations are generated using the Broad Parameter Subset in place of the Performance Parameter Subset. This may be because the reduced correlation in the predictions output by the different $MPM_1$s in the Broad Parameter Subset outweigh the increase in the variability of the risking performance of their MPMs.

The most marked difference in the rows of Table 7.2 is the huge reduction in the performance of the $\chi^2$ Statistic seen on moving from simulation parameter set $k_{19P}$ to set $k_{19P}$. This
loss of performance is likely due to the reduction in the quality of care variability associated with this change. That a similar loss of benchmarking performance is not seen for the SLL suggests that the SLL merely measures the risking performance of MPMs that happens to also be correlated with their benchmarking performance. It is also notable that, even with simulation parameter set $k^{19p}$ having only a mere 20% of the unaccounted for inter-ICU mortality rate variability as being due to differences in ICU quality of care, the SMR still greatly outperforms the SLL with 23% vs 40% error. This suggests that the APACHE Database lies well inside the regime in which there are sufficient ICUs and sufficient ICU quality of care variability for the $\chi^2$ Statistic to outperform the SLL and AUC in terms of its ability to discriminate between MPMs for use in ICU benchmarking.

The most important trends in Table 7.3 relate to benchmarking performance. The second row indicates that selecting an MPM at random from the high benchmarking performance set provides a correlation between the predicted and actual ICU quality of care of approximately 50%, while choosing the optimal MPM only increasing this value by approximately 3 percentage points. By contrast, the use of a random MPM from the set over the Null model

<table>
<thead>
<tr>
<th>MPM Selection Approach</th>
<th>Risking</th>
<th>$Z^{k^{19p}}$</th>
<th>% Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Internal SMR</td>
<td>Internal PQM</td>
</tr>
<tr>
<td>Null</td>
<td>-</td>
<td>13.9</td>
<td>13.9</td>
</tr>
<tr>
<td>Average</td>
<td>70.9</td>
<td>23.8</td>
<td>24.1</td>
</tr>
<tr>
<td>Best</td>
<td>75.5</td>
<td>25.5</td>
<td>25.8</td>
</tr>
<tr>
<td>HL</td>
<td>72.6</td>
<td>24.0</td>
<td>24.2</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>72.9</td>
<td>25.1</td>
<td>25.4</td>
</tr>
<tr>
<td>PVM</td>
<td>73.0</td>
<td>25.1</td>
<td>25.4</td>
</tr>
<tr>
<td>Other</td>
<td>75.5</td>
<td>24.4</td>
<td>24.6</td>
</tr>
</tbody>
</table>

Table 7.4 Table to display the mean performance of the MPMs selected by the given selection approach relative to the tasks defined in Subsection 7.3.1. Simulation parameter set $k^{19p}$ was used to generate the simulations from which these results were derived. The SMR (Standardized Mortality Ratio) and PQM (Posterior Quality Mean) indicate which ICU evaluator was implemented when performing ICU benchmarking.
### Table 7.5

Table to display the mean performance of the MPMs selected by the given selection approach relative to the tasks defined in Subsection 7.3.1. Simulation parameter set $k^{4B}$ was used to generate the simulations from which these results were derived. The SMR (Standardized Mortality Ratio) and PQM (Posterior Quality Mean) indicate which ICU evaluator was implemented when performing ICU benchmarking.

<table>
<thead>
<tr>
<th>MPM Selection Approach</th>
<th>Risking</th>
<th>$Z_k^{4B}$ / % Correlation Benchmarking</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Internal SMR</td>
<td>Internal PQM</td>
</tr>
<tr>
<td>Null</td>
<td>-</td>
<td>28.3</td>
<td>28.3</td>
</tr>
<tr>
<td>Average</td>
<td>58.8</td>
<td>41.4</td>
<td>41.6</td>
</tr>
<tr>
<td>Best</td>
<td>65.4</td>
<td>45.0</td>
<td>45.3</td>
</tr>
<tr>
<td>HL</td>
<td>62.6</td>
<td>42.1</td>
<td>42.4</td>
</tr>
<tr>
<td>$\chi^2$</td>
<td>61.1</td>
<td>44.6</td>
<td>45.0</td>
</tr>
<tr>
<td>PVM</td>
<td>61.4</td>
<td>44.7</td>
<td>45.0</td>
</tr>
<tr>
<td>Other</td>
<td>65.4</td>
<td>42.7</td>
<td>42.9</td>
</tr>
</tbody>
</table>

Increases the correlation by over 20 percentage points. The somewhat modest 3 percentage point upper limit on the improvement that can expected from selecting better benchmarking MPMs is likely due to some combination of the MPMs in the Performance $MPMI$ Subset being very similar to one another and the fact that the reported value is for selecting the best of only two MPMs.

Another clear trend from Table 7.3 is that the MPM evaluators in the high risking performance group are only able to exploit less than one half of this potential 3 percentage point gain, whereas the MPM evaluators in the high benchmarking performance group are able to exploit almost all of it. This approximate doubling of the increase in performance from 1.5 to 3 percentage points supports the potential for transitioning from the use of the AUC to using either the $\chi^2$ or PVM when assessing an MPM’s ICU benchmarking performance. It is also interesting that for both internal and external benchmarking, the performance of both the $\chi^2$ and PVM with the PQM exceeds the performance of the ideal selection approach with the SMR.
Quartering the proportion of inter-ICU variability that is assumed to be accounted for by differences in ICU Quality of Care (as corresponds to shifting from Table 7.3 to Table 7.4) results in an approximate halving of all benchmarking performance measures as well as a modest reduction in risking performance measures. Meanwhile, moving from the Performance $MPMI^1$ Subset to the Broad $MPMI^1$ Subset (as corresponds to shifting from Table 7.3 to Table 7.5) reduces both risking and benchmarking performance, in line with expectations.
Chapter 8

Partitioning and the Assessment of Generalization Error

The generalization error of an MPM is a measure of the performance that it is expected to exhibit when applied to new patients. Such measures are potentially useful for MPM performance comparisons and in determining if a given MPM is sufficiently accurate for a given task. In Chapter 5 a total of four MPMC stages were introduced that focused on the assessment of generalization errors:

- Partitioning matrix inducers (Section 5.2).
- MPM evaluators (Section 5.4).
- Performance point estimate inducers (Section 5.5).
- Standard error estimate inducers (Section 5.6).

In the present Chapter the term ‘partitioning approach’ is used to refer to both the generation of a partitioning matrix through the application of a partitioning matrix inducer and the use of such a partitioning matrix to partition patients into disjoint training and test sets. Some partitioning approaches split patients into multiple training and test sets that can give rise
to more than one application of an MPM evaluator. For such partitioning approaches, a performance point estimate inducer specifies how such MPM evaluations are combined in order to output a single performance point estimate, while a standard error estimate inducer provides a method for quantifying the degree of uncertainty associated with such a performance point estimate.

For convenience a ‘party’ is introduced as a unique combination of a partitioning matrix inducer, a performance point estimate inducer and a standard error estimate inducer. The notation $\alpha-\beta-\gamma$ is used to refer to the party consisting of the $\alpha$’th partitioning matrix inducer, the $\beta$’th performance point estimate inducer and the $\gamma$’th standard error estimate inducer. A complete list of the parties used in this Chapter is provided in Table 8.1. The inclusion of a ‘⋆’ in a party’s specification indicates a set of parties that includes all possible values for the index that is replaced by the ⋆. For example the notation 3-1-⋆ would refer to all of 3-1-1, 3-1-2 and 3-1-3.

The focus of this Chapter is on investigating which parties are the most appropriate for inclusion in the seven types of MPMCs listed in Section 1.1. The structure of this Chapter is as follows: Section 8.1 details the potential issues associated with Patient Holdout - the party used in the Standard MPMC. Section 8.2 examines the traits that a good party should possess. Section 8.3 concretely defines how to measure the different aspects of a party’s performance. The remainder of this Chapter then evaluates and discusses the performance of different parties using both the Simulator and the APACHE Database.

## 8.1 Issues with Patient Holdout

Patient Holdout is the partitioning approach implemented in the Standard MPMC, and involves randomly partitioning patients into a single training set and a single test set. Each $MPM_1$ is then applied in turn to the training set to generate a total of $M$ MPMs. Each MPM is then applied to the patients in the test set and the output predictions evaluated using
### Table 8.1

<table>
<thead>
<tr>
<th>Partitioning matrix inducer</th>
<th>Performance point estimate inducer</th>
<th>Set comparison</th>
<th>Individual bootstrapping</th>
<th>ICU bootstrapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%-patient holdout</td>
<td>-</td>
<td>-</td>
<td>1-⋆-2</td>
<td>1-⋆-3</td>
</tr>
<tr>
<td>4-fold patient cross validation</td>
<td>-</td>
<td>2-⋆-1</td>
<td>2-⋆-2</td>
<td>2-⋆-3</td>
</tr>
<tr>
<td>10-repate 2-fold patient cross-validation</td>
<td>Mean first</td>
<td>3-1-1</td>
<td>3-1-2</td>
<td>3-1-3</td>
</tr>
<tr>
<td>10-repate 2-fold patient cross-validation</td>
<td>Mean second</td>
<td>3-2-1</td>
<td>3-2-2</td>
<td>3-2-3</td>
</tr>
<tr>
<td>60%-ICU holdout</td>
<td>-</td>
<td>-</td>
<td>4-⋆-1</td>
<td>4-⋆-3</td>
</tr>
<tr>
<td>4-fold ICU cross validation</td>
<td>-</td>
<td>5-⋆-1</td>
<td>5-⋆-2</td>
<td>5-⋆-3</td>
</tr>
<tr>
<td>10-repate 2-fold ICU cross-validation</td>
<td>Mean first</td>
<td>6-1-1</td>
<td>6-1-2</td>
<td>6-1-3</td>
</tr>
<tr>
<td>10-repate 2-fold ICU cross-validation</td>
<td>Mean second</td>
<td>6-2-1</td>
<td>6-2-2</td>
<td>6-2-3</td>
</tr>
</tbody>
</table>

Table 8.1 Table containing all of the parties considered in this Chapter. The first and second columns specify the partitioning matrix inducer and performance point estimate inducer that a given party incorporates. The location of the party within the final three columns specifies the standard error estimate inducer that the given party incorporates. The labels by which the various parties are referred to are provided in the final three columns of the table. A ‘-’ indicates a missing entry because the given combination of partitioning matrix inducer, performance point estimate inducer and standard error estimate inducer does not form a valid party. The inclusion of a ‘⋆’ in a party’s specification indicates that the outputs of the party are independent of the performance point estimate inducer that it includes.
8.1 Issues with Patient Holdout

an MPM evaluator. Patient Holdout provides a very direct measure of the output MPM’s generalization error and makes it much easier for researchers to iteratively update models without over-fitting. They may simply constrain themselves to only look at the patients in the training set while constructing their models and then evaluate the final model on the unseen test set.

That Patient Holdout splits on patients rather than ICUs indicates that the performance point estimates that it outputs will be appropriate for use internally, i.e. on new patients who are assigned to the same ICUs as those to which the MPMC incorporating Patient Holdout was applied. However, such estimates may not be appropriate for use externally, i.e. on new patients assigned to ICUs from outside of those in the ICU Database to which the MPMC was applied. For ICU benchmarking, Patient Holdout may also invalidate the assumption that MPMs cannot account for ICU quality of care, potentially resulting in overly optimistic generalization error estimates.

In Patient Holdout, only a fraction of the patient records are used to train the output MPM, suggesting that a superior MPM could be trained using all of the patient records. Similarly, the use of only a fraction of the patient records when assessing the MPM’s performance suggests that such an estimate might be unnecessarily unstable. Patient Holdout will therefore tend to output both a suboptimal MPM and a suboptimal estimate of the generalization error associated with this final MPM.

The Standard MPMC does not need to specify a performance point estimate inducer because Patient Holdout incorporates only a single test set. However, the absence of a standard error estimate inducer in the Standard MPMC implies that it does not output a standard error estimate. This then makes it difficult to know the extent to which the output performance point estimate can be relied upon. The solution then is to add a standard error estimate inducer to the Standard MPMC using one of the top two parties in Table 8.1.
8.2 Measuring Party Performance

Having established that the party used in the Standard MPMC is suboptimal (Section 8.1) and provided various alternatives (Table 8.1), it is now necessary to determine which of these alternatives is the most appropriate for each of the various types of MPMC listed in Section 1.1. To achieve this, suppose that one has an MPMC that includes a given party and an $MPM^2$ containing a single $MPM^1$, and that can be applied to produce the following outputs:

1. An MPM.

2. A performance point estimate.

3. A standard error estimate.

4. A quality of care estimate per ICU.

Suppose that one also has a group of patients that have been partitioned into two groups: a large patient subset that contains the vast majority of patients and a small patient subset that contains the remaining patients. The MPMC could then be applied to the small patient subset in order to generate the above outputs. The output MPM could then be applied to the patients in the large patient subset and the output risk estimates evaluated in order to provide a measure of the actual generalization error of the output MPM. A small measure for the generalization error is then an indication of a high performing MPM. A performance point estimate with a value similar to that of the generalization error indicates a good performance point estimate, and a standard error estimate with a value similar to the absolute value of the difference between the generalization error and the performance point estimate indicates a good standard error estimate.

A similar approach can be used to assess parties for MPMCs that compare multiple $MPM^1$s. When an MPMC compares two $MPM^1$s it outputs a performance point estimate
for each and is effectively stating that the $MPMI^1$ associated with the largest performance point estimate is the best. The truth of such an assertion can then be checked by determining which of the two output MPMs is associated with the greatest generalization error. For an MPMC the outputs of which are intended for use in risking (benchmarking), the above should be performed using an MPMC that applies an MPM evaluator that is appropriate for risking (benchmarking). For an MPMC the outputs of which are intended for use internally (externally), the partitioning of the patients into the large patient and small patient subsets should be performed over patients (ICUs).

One way to generate the required patient subsets is to apply the Simulator with a large number of blocks and take the small patient subset to be a single block and the large patient subset to contain the patients from all other blocks. Another approach is to take the entire APACHE Database and partition it into blocks in a similar manner to an application of the Simulator, except that the partitioning needs be over patients rather than ICUs because of the limited number of ICUs in the APACHE Database. Given such a partitioning of the APACHE Database, one could again take the small patient subset to be a single block and the large patient subset to contain the patients from all other blocks. In this Chapter, the two above approaches are used to assess the seven types of MPMCs listed in Section 1.1 as follows:

1. Internal Risking: The APACHE Database is used because it allows parties to be tested on an actual ICU Database with genuine $MPMI^1$s.

2. External Risking: The Simulator is used because the APACHE Database contains too few ICUs to be partitioned appropriately.

3. External Benchmarking: The Simulator is used because actual quality of care values are required to evaluate quality of care estimates.

4. Internal Benchmarking: The Simulator is used because actual quality of care values are required to evaluate quality of care estimates.
5. Internal Risking Comparison: The APACHE Database is used because it allows parties to be tested on an actual ICU Database with genuine MPMI's.

6. External Risking Comparison: The Simulator is used because there would be too few ICUs in the small patient subset if the APACHE Database was used.

7. Benchmarking Comparison: The Simulator is used because actual quality of care values are required to evaluate quality of care estimates.

Exact definitions for the various aspects of party performance are provided in the next Section. The methods used to evaluate these performance measures are provided in Section 8.4. In Chapter 7 it was determined that the SLL and $\chi^2$ values embody the risking performance and benchmarking performance, respectively, of MPMs. As such the present Chapter will focus on generalization errors in terms of the SLL for risking and the $\chi^2$ for benchmarking. Because some parties make predictions for only a subset of ICUs it is necessary to ensure that the $\chi^2$ reported in this Chapter are robust to changes in ICU numbers. To ensure this, all such $\chi^2$ values are scaled by the number of ICUs over which they are evaluated.

### 8.3 Defining Party Performance

**Risk Array Generation**

Let $\alpha \in \{1, \ldots, 22\}$ index the 22 parties listed in Table 8.1. Suppose that one has applied the Simulator with $B = 20$ blocks, $M = 2 \text{ MPMI}^1$'s, simulation parameter set $k^4P$ and with the partitioning matrix inducer corresponding to the $\alpha$’th party in order to output:

1. An $I \times 20$ response matrix, $y$, for which $b_{yi}$ gives the response for the $i$'th patient in the $b$’th block.

2. A length $I$ membership vector, $h$. 
3. A $K \times 20$ ICU quality of care matrix, $\mathbf{q}$, for which $b q_k$ gives the quality of care provided by the $k$'th ICU in the $b$'th block.

4. An $I \times A \times 2 \times 20 \times 20$ risk array, $\mathbf{p}$, for which $b \hat{p}_{ia}^m$ gives:

(a) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th MPMI$^1$ evaluated on the $a$'th partition if $\beta = b$.

(b) The estimated risk for the $i$'th patient in the $b$'th block generated by the $m$'th final MPMI$^1$ output on the $\beta$'th block if $\beta \neq b$.

Or alternatively suppose that one has partitioned the patients in the APACHE Database into 20 blocks of equal size without regard to their membership, in order to generate the same outputs as above except without an ICU quality of care matrix, and for $\alpha \in \{1, \ldots, 11\}$ corresponding to the top 11 entries in Table 8.1. Set $\gamma = 22$ if outputs were from the Simulator, or $\gamma = 11$ if from the APACHE Database.

**Performance Estimation**

Suppose that the following four sets of values have been generated from the above outputs:

1. A $20 \times \gamma \times 2 \times 2$ array of performance point estimates, $\hat{\mathbf{U}}$, for which $b \alpha \hat{U}_{aw}^m$ represents the performance point estimate associated with the $w$'th MPM evaluator, output by the application of the performance point estimate inducer associated with the $\alpha$'th party to the $b \hat{p}_{ia}^m$ output by applying the partitioning approach associated with the $\alpha$'th party.

2. A $20 \times \gamma \times 2 \times 2$ array of standard error estimates, $\hat{\mathbf{V}}$, for which $b \alpha \hat{V}_{aw}^m$ represents the standard error estimate associated with the $w$'th MPM evaluator, output by the application of the standard error estimate inducer associated with the $\alpha$'th party to the $b \hat{p}_{ia}^m$ output by applying the partitioning approach associated with the $\alpha$'th party.
3. A $20 \times \gamma \times 2 \times K$ array of quality of care estimates, $\hat{q}$, for which $\frac{b}{\alpha} \hat{q}_{m}^{k}$ represents the quality of care estimate output for the $k'$th ICU by the application of the SMR to the $\frac{b}{\beta} \hat{p}_{m}^{\beta}$ output by applying the partitioning approach associated with the $\alpha'$th party.

4. A $20 \times 20 \times \gamma \times 2 \times 2$ array of performance evaluations, $U$, for which $\frac{b}{\beta} U_{w}^{\gamma}$ represents the output from applying the $w$'th MPM evaluator to the $\frac{b}{\beta} \hat{p}_{m}^{\beta}$ output by applying the partitioning approach associated with the $\alpha'$th party.

Evaluating Final MPM Performance

Let $\frac{b}{\alpha} Y_{m}$ refer to the MPM that output the $\frac{b}{\beta} \hat{p}_{m}^{\beta}$ for all $\beta \neq b$. Then the mean actual performance of $\frac{b}{\alpha} Y_{m}$ in terms of the $w$'th MPM evaluator is taken to be:

$$\frac{b}{\alpha} U_{m}^{w} = E_{\beta \neq b} (\frac{b}{\beta} U_{m}^{w})$$  \hspace{1cm} (8.1)$$

For simulations the mean actual performance of the MPMs output using the partitioning approach associated with the $\alpha'$th party, in terms of the $w$'th MPM evaluator is then:

$$\alpha U_{*}^{w} = E_{m} (E_{b} (\frac{b}{\alpha} U_{m}^{w}))$$  \hspace{1cm} (8.2)$$

If instead the APACHE Database was used, then the two $MPMI^1$s implemented by the MPMC will differ in a consistent manner, and two values for the mean actual performance are required as:

$$\alpha U_{m}^{w} = E_{b} (\frac{b}{\alpha} U_{m}^{w})$$  \hspace{1cm} (8.3)$$
8.3 Defining Party Performance

**Evaluating Performance Point Estimates**

The error in the performance point estimate output for \( b \gamma_m \) in terms of the \( w \)'th MPM evaluator is:

\[
\beta R_m^w = \beta \hat{U}_m^w - \alpha U_m^w.
\]  (8.4)

**Evaluating Standard Error Estimates**

The mean actual standard error associated with the performance point estimates output by the MPMC in terms of the \( w \)'th MPM evaluator for simulations is then:

\[
\alpha V^w = \sqrt{E_b(E_m(\beta R_m^w)^2)}
\]  (8.5)

If instead the APACHE Database was used, then the two \( MPMI^1 \)s implemented by the MPMC will differ in a consistent manner, and two values for the mean actual standard error are required as:

\[
\alpha V^w = \sqrt{E_b(\beta R_m^w)^2}
\]  (8.6)

**Evaluating MPM Discrimination**

The extent to which the \( \alpha \)'th party can discriminate between MPMs when using the \( w \)'th MPM evaluator is taken to be given by:

\[
a D^w = \kappa(b \hat{U}_2^w - \alpha U_2^w, b \hat{U}_1^w - \alpha U_1^w)
\]  (8.7)

where \( \kappa(\zeta, \eta) \) denotes the Kendall correlation between vectors \( \zeta \) and \( \eta \).
8.4 Evaluating Party Performance

Evaluating ICU Discrimination

The performance of the ICU quality of care estimates for the \( m \)'th MPM on the \( b \)'th block and generated by the \( \alpha \)'th party is taken to be:

\[
_{\alpha}bS_m = \kappa^{(b\hat{q}_m, b\hat{q})},
\]  

(8.8)

with the mean ICU discrimination offered by the \( \alpha \)'th party is taken to be:

\[
_{\alpha}S = E_b(E_m(b_{\alpha}S_m)).
\]  

(8.9)

8.4 Evaluating Party Performance

Let \( \alpha \in \{1, \ldots, 22\} \) index the 22 parties listed in Table 8.1 ordered from top (left) to bottom (right). Further let \( w = 1 \) and \( w = 2 \) index the SLL and the \( \chi^2 \) Statistic, respectively. Then the measures of party performance defined above are evaluated using the Simulator and the APACHE Database as described below.

Simulations

1. For all combinations of \( \alpha \in \{1, \ldots, 22\} \) and \( \pi \in \{1, 2, \ldots, 100\} \):

   (a) Apply the Simulator with \( B = 20 \) blocks, \( M = 2 \) MPM1's, simulation parameter set \( k^{4P} \) and the partitioning matrix inducer corresponding to the \( \alpha \)'th party.

   (b) For all combinations of \( w \in \{1, 2\}, b \in \{1, 2, \ldots, 20\} \) and \( m \in \{1, 2\} \) calculate and retain the \( _{\alpha}bR_m, _{\alpha}\hat{V}_m, _{\alpha}U_{\alpha}^m, _{\alpha}V_{\alpha}^w, _{\alpha}D_{\alpha}^w \) and \( _{\alpha}bS_m \).

2. Output each of the \( _{\alpha}bR_m \) and \( _{\alpha}\hat{V}_m \) values that were calculated.
3. Output each element of the $\alpha U_w^\star$, $\alpha V_w^\star$, $\alpha D_w^\star$ and $b S_m^\star$ as the mean of that element averaged over $\pi \in \{1, 2, \ldots, 100\}$.

The APACHE Database

1. The patients in the APACHE Database are randomly split to form $B = 20$ blocks of 11,550 patients.

2. For $\alpha \in \{1, \ldots, 11\}$:
   
   (a) An MPMC is created by combining the $\alpha$’th partitioning approach with the set of MPMCs defined in Table 8.2.

   (b) For $b \in \{1, \ldots, 20\}$: The MPMC is applied to the patients in the $b$’th block and the output MPMs applied to the patient profiles of patients not in the $b$’th block. This step should then provide the same outputs as an application of the Simulator except without a quality of care matrix, $q$.

   (c) For all combinations of $b \in \{1, 2, \cdots, 20\}$ and $m \in \{1, 2\}$ calculate and output the $b \alpha R_m^1$, $b \hat{V}_m^1$, $\alpha U_m^1$, $\alpha V_m^1$ and $\alpha D^1$.

8.5 Results

Performance Point Estimation

The distribution of errors associated with the performance point estimates for different parties are provided in Figures 8.1, 8.2 and 8.3, for each of risking on simulations, benchmarking on simulations and risking on the APACHE Database, respectively. Ideally, the median error (thick horizontal line) for a given party should fall as close to zero (red line) as possible. A median error with a positive value indicates an overly pessimistic (large) estimate of
Table 8.2 Specification for a set of MPMCs used to assess different parties. The External Partitioning Matrix Inducer is not defined, with this option depending on the party with which the MPMC is applied. Descriptions of the various steps are linked through Table 5.4.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td></td>
<td>Response</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>Mortality at Hospital Discharge</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Prediction Time Inducer</td>
</tr>
<tr>
<td></td>
<td>2.1</td>
<td>Deterministic Variable</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Treatment Event Retention</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>Treatment Events Retained</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Transfer Event Obfuscation</td>
</tr>
<tr>
<td></td>
<td>4.0</td>
<td>Transfer Events Obfuscated</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Unconditional Stay Exclusions</td>
</tr>
<tr>
<td></td>
<td>5.2</td>
<td>Standard Exclusions</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Conditional Stay Exclusions</td>
</tr>
<tr>
<td></td>
<td>6.1</td>
<td>First Stay Only</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Preprocessor Inducer</td>
</tr>
<tr>
<td></td>
<td>7.0</td>
<td>Standard</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Internal Partitioning Matrix</td>
</tr>
<tr>
<td></td>
<td>8.0</td>
<td>4-Fold Patient Holdout</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>External Partitioning Matrix Inducer</td>
</tr>
<tr>
<td></td>
<td>9.*</td>
<td>Undefined</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>Internal MPM Evaluator</td>
</tr>
<tr>
<td></td>
<td>10.0</td>
<td>Standardized Log-Likelihood</td>
</tr>
<tr>
<td>Option</td>
<td></td>
<td>MPMI²</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>{{RLR},{\NN_1}}</td>
</tr>
</tbody>
</table>

Confidence Interval Estimation

The distribution of the standard error estimates for the various parties are provided in Figures 8.4, 8.5 and 8.6, for each of risking on simulations, benchmarking on simulations and risking on the APACHE Database, respectively. Ideally the median standard error estimate (thick horizontal line) for a given party should fall as close to the actual standard error (red line) as possible. A median standard error estimate above the red line indicates an overly pessimistic (large) estimate of the uncertainty in the associated performance point estimate, while a median standard error estimate below the red line indicates an overly optimistic (small) estimate. Generally, a median standard error estimate that is too large is preferred over one that is too small, as overly optimistic estimates are generally more dangerous than overly
Fig. 8.1 Modified box-plot summarizing the distribution of Standardized Log-Likelihood (SLL) performance point estimate errors for different parties applied to simulated data. Each box is constructed as follows: All values of $\alpha R^1$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents no difference between the performance point estimates and the actual generalization error.
Fig. 8.2 Modified box-plot summarizing the distribution of $\chi^2$ performance point estimate errors for different parties applied to simulated data. Each box is constructed as follows: All values of $\alpha R^2$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents no difference between the performance point estimates and the actual generalization error.
Fig. 8.3 Modified box-plots summarizing the distribution of Standardized Log-Likelihood (SLL) performance point estimate errors for different parties applied to the APACHE Database. The two panels correspond to different learner inducers, with the left panel representing Regularized Logistic Regression and the right panel representing Nearest Neighbors. Each box in the left panel is constructed as follows: All values of $\alpha R_1^1$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents no difference between the performance point estimates and the actual generalization error. The right panel is constructed similarly, except that $\alpha R_1^1$ is replaced with $\alpha R_1^2$. 
pessimistic ones. Ideally the spread of standard error estimates (the size of a given box) for a
given party should be as small as possible. A red line with a small value indicates that the
associated party provides accurate performance point estimates. For brevity, only the parties
which did not exhibit a large performance point estimate bias have been included in these
Figures.

Further Aspects of Party Performance

In Section 8.2, several aspects of party performance beyond the accuracy of performance
point estimates and confidence interval estimates were considered. Each of these further
measures of party performance are quantified in Table 8.3 for every valid combination
of partitioning matrix inducer and performance point estimator. The second through fifth
columns of the table provide the mean actual performance of the final MPMs output by
the different parties, with small values indicative of a superior final MPM. The sixth and
seventh columns provide the mean correlation between estimated and actual differences in
MPM performance, with a value of unity indicating perfect discrimination. That the values
in the sixth column are much closer to unity than those in the seventh column indicates
that discriminating between MPMs for use in benchmarking is much more difficult than
discriminating between MPMs for use in risking, in line with the findings of Chapter 7. The
final column provides the mean correlation between the QoC estimates output by the given
party and the true QoC values, with larger values indicating parties that are better at assessing
the performance of the ICUs in the ICU Database to which the associated MPMC is applied.
8.5 Results

<table>
<thead>
<tr>
<th>Party</th>
<th>Output MPM Performance</th>
<th>MPM Discrimination</th>
<th>$S$ (ICU Discrimination)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U^1$ (SLL)</td>
<td>$U^2$ ($\chi^2$)</td>
<td>$D^1$ (SLL)</td>
</tr>
<tr>
<td>1-***</td>
<td>0.2410</td>
<td>0.3075</td>
<td>11.0</td>
</tr>
<tr>
<td>4-***</td>
<td>0.2411</td>
<td>-</td>
<td>11.1</td>
</tr>
<tr>
<td>2-***</td>
<td>0.2410</td>
<td>0.2752</td>
<td>0.3053</td>
</tr>
<tr>
<td>5-***</td>
<td>0.2410</td>
<td>-</td>
<td>11.0</td>
</tr>
<tr>
<td>3-1-**</td>
<td>0.2410</td>
<td>0.2752</td>
<td>0.3053</td>
</tr>
<tr>
<td>6-1-**</td>
<td>0.2410</td>
<td>-</td>
<td>11.0</td>
</tr>
<tr>
<td>3-2-**</td>
<td>0.2410</td>
<td>0.2752</td>
<td>0.3053</td>
</tr>
<tr>
<td>6-2-**</td>
<td>0.2410</td>
<td>-</td>
<td>11.0</td>
</tr>
</tbody>
</table>

Table 8.3 Table summarizing aspects of party performance outside of the accuracy of their associated performance point estimates and confidence interval estimates. The top row that is split into three levels provides descriptions for each column. The first two levels provide details of the measure of party performance that is contained within a given column. For the second, third and fourth columns the performance measure is the SLL of the MPM that is output by the given party. For the fifth column the performance measure is the $\chi^2$ of the MPM that is output by the given party. For the sixth (seventh) column the performance measure is the ability of the MPMC to identify which of its two MPMs offers the best SLL ($\chi^2$) when applied to new patients from ICUs outside of those to which the MPMC was applied. The final column provides the mean correlation between the QoC estimates output by the given party and the true QoC values. The third level of the top row specifies the combination of $MPMI^1$ and patients to which the party is applied. The three options are ‘Sim’ for which both the $MPMI^1$ and patients are those output by the Simulator (Chapter 6), ‘RLR’ for which Regularized Logistic Regression was applied to the APACHE Database and ‘NN’ for which Nearest Neighbors was applied to the APACHE Database.
Fig. 8.4 Modified box-plot summarizing the distribution of SLL standard error estimates output by the different parties applied to simulated data. Each box is constructed as follows: All values of $\hat{\alpha V^1}$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents the actual standard error ($\alpha V^1_*$) associated with the performance point estimates of the given party.
Fig. 8.5 Modified box-plot summarizing the distribution of $\chi^2$ standard error estimates output by the different parties applied to simulated data. Each box is constructed as follows: All values of $\hat{\chi}^2$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents the actual standard error ($\alpha V^2_e$) associated with the performance point estimates of the given party.
Fig. 8.6 Modified box-plots summarizing the distribution of SLL standard error estimates output by the different parties applied to the APACHE Database. The two panels correspond to different learner inducers, with the left panel representing Regularized Logistic Regression and the right panel representing Nearest Neighbors. Each box in the left panel is constructed as follows: All values of $\hat{\alpha}$ for which $\alpha$ corresponds to the given set of parties (as specified along the bottom axis of the plot) are collapsed into a single vector. The thick horizontal black line gives the median of this vector. The thin horizontal black lines bound the 2nd and 3rd quartiles and the whiskers bound a balanced 95% confidence interval for this vector. The horizontal red line represents the actual standard error ($\alpha V_1$) associated with the performance point estimates of the given party. The right panel is constructed similarly, except that $\hat{\alpha} V_1$ and $\alpha V_1$ are replaced with $\hat{\alpha} V_2$ and $\alpha V_2$, respectively.
8.6 Discussion

Final MPM Performance

For MPMCs that are to be used for external benchmarking or any form of risk scoring, the performance of the output MPM is of prime importance. Four measures of how the different parties affect MPM performance are displayed in columns 2-5 of Table 8.3. All values within a given column are identical, except for those associated with Holdout (1-⋆-⋆ and 4-⋆-⋆), because the final MPMs output by all such parties are identical.

On simulations (columns 2 and 5) there is very little difference between the final MPM performance of 1-⋆-⋆ and that of the non-Holdout parties. This is likely due to the large number of patients included in a single simulation block giving rise to a near optimal set of MPM parameters whether the $MPMI^1$ is applied to all patients in a given block (as is the case for non-Holdout parties) or if it is applied to just 60% of the patients in a given block (as is the case for Holdout). In contrast, the $MPMI^1$s applied to a single block from the APACHE Database receive only a twentieth as many patient profiles as those applied to a single simulation block. In such a scenario, an $MPMI^1$ that receives additional patient profiles may be able to fit MPM parameter values substantially better. This may then explain why on the APACHE Database (columns 3 and 4) the final MPM performance for 1-⋆-⋆ is considerably worse than that for the non-Holdout parties. This effect seems particularly pronounced for NN because the $MPMI^1$ is much less stable and therefore more likely to suffer from small training sets than is RLR.

The average performance of the MPMs output using 4-⋆-⋆ is worse than that of the other parties, especially with regard to benchmarking performance (column 5). This is to be expected, as the MPMs output using 4-⋆-⋆ are trained on patients from fewer ICUs than those trained using other parties, with the output MPM therefore more prone to instability due to differences in the mean case-mix and quality of care provided for the patients on which the final MPM was trained and those to which the final MPM is applied. In all, the differences
in performance of the final MPMs associated with different parties offer moderate evidence against the use of either 1-⋆-⋆ or 4-⋆-⋆.

**Performance Point Estimation**

The bias figures associated with the performance point estimates output by different parties on the APACHE Database were provided in Figure 8.3. When considering these distributions, it is important to note that they are generated from a single dataset, and therefore that their finer details may not generalize. However, there are two trends that are sufficiently pronounced as to warrant consideration. The first is that the distributions for 1-⋆-⋆ are much less concentrated than those for the other parties. This corroborates the earlier concern that 1-⋆-⋆ may provide inaccurate performance point estimates due to its excluding 60% of patients from any test set. This result then provides moderate evidence against the use of 1-⋆-⋆.

The second trend is that the difference in bias between 3-1-⋆ and 3-2-⋆ is much smaller for RLR than it is for NN. The likely cause is that 3-1-⋆ is effectively applying bagging in order to generate more optimistic performance point estimates, with this effect more pronounced for NN because it outputs MPMs that are much less stable than those output by RLR. This trend indicates that one cannot rely on the performance point estimate output by 3-1-⋆, and suggests that it should be removed from further consideration with regard to internal risking.

The bias figures associated with the performance point estimates output by different parties on simulated data were provided in Figures 8.1 and 8.2. From Figure 8.2, it is immediately obvious that the parties that split on individuals (1-⋆-⋆, 2-⋆-⋆, 3-1-⋆ and 3-2-⋆) provide benchmarking performance point estimates that are massively biased. The over-optimism inherent in these estimates likely reflects the concern that originally motivated this Chapter: that splitting patients from a single ICU between training and test sets invalidates the assump-
tion that MPMs cannot account for case-mix. The same effect is also apparent, although to a greatly diminished extent, when considering risking performance point estimates (Figure 8.1). This property of consistent over-optimism is deemed sufficiently undesirable as to warrant the removal of these parties from further consideration with regard to both external risking and external benchmarking.

The causes of the mildly over-optimistic predictions output by 6-1-\( \ast \) and the much greater variance associated with 4-\( \ast \)-\( \ast \) are likely equivalent to those discussed earlier for 3-1-\( \ast \) and 1-\( \ast \)-\( \ast \), respectively. For external risking and external benchmarking this then leaves 4-\( \ast \)-\( \ast \), 5-\( \ast \)-\( \ast \) and 6-2-\( \ast \) as the sets of parties that warrant further investigation.

**Standard Error Estimation**

The distributions of the standard error estimates output by the different parties on the APACHE Database were provided in Figure 8.6. Again, these distributions are generated from a single dataset and therefore only the most pronounced trends should be considered. It is quite clear that 1-\( \ast \)-2 should be ruled out because of its very large standard error (red line) while 2-\( \ast \)-1 should be ruled out because of how variable its standard error estimates are. It seems there may not be sufficient information in Figure 8.6 to reliably choose which of 2-\( \ast \)-2, 3-2-1 and 3-2-2 is most appropriate for providing standard error estimates for internal risking. However, 3-2-1 should perhaps be favored over 2-\( \ast \)-2 and 3-2-2 because it is the only party still under consideration that should be able to account for both variability in training sets and variability in test sets, and is therefore likely to be the safest of the options.

The distributions of the standard error estimates output by the different parties on the simulated data were provided in Figures 8.4 and 8.5. It is immediately obvious from both plots that all of the methods involving bootstrapping on patients (4-\( \ast \)-2, 5-\( \ast \)-2 and 6-2-2) are associated with massively over-optimistic standard error estimates. The best party overall, in terms of both minimum bias and minimum variability, appears to be 6-2-1.
Discrimination

In terms of MPM discrimination (columns 6 and 7 in Table 8.3), 4-⋆-⋆ is by far the worst set of parties, with the likely cause again being that it only uses 40% of ICUs when evaluating MPM performance. Of the remaining parties, the value of 0.926 offered by 5-⋆-⋆ for the $\chi^2$ discrimination is quite poor while the remainder of methods are quite competitive. The poor performance of 5-⋆-⋆ (relative to 6-1-⋆ and 6-2-⋆) may be due to each prediction being made only once per patient, such that model instability is not properly mitigated.

In terms of ICU discrimination (column 8 in Table 8.3) 6-2-⋆ again performs well while 1-⋆-⋆ is the only set of parties that performs extremely poorly. The poor ICU discrimination offered by 1-⋆-⋆ is likely due to the inclusion of only a fraction of the patients in each ICU when calculating the SMR\(^1\). It is also notable that 4-⋆-⋆ can only generate ICU evaluations for the 40% of ICUs that are in the test set. This then makes 4-⋆-⋆ inappropriate for internal ICU performance evaluation in spite of its large ICU discrimination value.

8.7 Conclusions

In the previous Section, it was confirmed that failing to ensure that the patients from a single ICU are contained wholly within either the training set or test set every time an MPM1 is applied results in highly over-optimistic benchmarking performance point estimates. It was also confirmed that the performance point estimates associated with Holdout are far less stable than those associated with parties that implement cross-validation. Simulations were used to show that the best overall party was 6-2-1, which demonstrated good performance over each of the following tasks:

- External Risking

\(^1\)This poor performance is not an issue for the APACHE IV model as it was designed for use in external benchmarking rather than internal benchmarking.
8.7 Conclusions

- External Benchmarking
- Internal Benchmarking
- External Risking Comparison
- Benchmarking Comparison

The APACHE Database was then used to show that 3-2-1 demonstrated the best performance for each of:

- Internal Risking
- Internal Risking Comparison
Chapter 9

Improving Internal Risking

Chapters 7 and 8 focused on modifying the Standard MPMC such that it provides better generalization error estimates with which to select the most appropriate MPMs. The present Chapter focuses on how to alter the Standard MPMC such that it outputs more appropriate MPMs for use in internal risking. The two MPM-related deficiencies that are addressed are the potential lack of predictive accuracy and the lack of continuous predictions.

Predictive Accuracy

The relative accuracy of mortality predictions made by each of the APACHE IV Model and nurses was discussed in Subsection 2.2.3, with the conclusion that it was unclear which of the two made better predictions. In a high-resource clinical environment such as an ICU, this similarity limits the potential use of MPMs in deterioration alarms: the new information with which predictions could be updated only becomes available as an ICU nurse records an event, and at such times the nurse will generally notice and alert others to abnormal events that are indicative of patient deterioration without the need for reference to an MPM. In such a scenario, the main utility of a deterioration alarm of moderate accuracy is as a backup that alerts physicians to clear cases of deterioration that have been missed, potentially due to human error or staff inexperience. In contrast, a highly accurate MPM could provide better
alerts than could be generated by ICU nurses, and could therefore improve care in even the most well-resourced of ICUs. In this Chapter, a variety of potential changes to the Standard MPMC are considered which could improve the accuracy of output MPMs, with the primary focus on more advanced machine learning techniques and more prolific feature extractor inducers.

**Continuous Predictions**

For deterioration alarms, it is clearly the case that a single prediction made 24 hours after ICU admission is insufficient. Instead, predictions should be updated as soon as additional information becomes available in the form of new events. One simple way to achieve this is to use a Standard MPM such as the APACHE IV model and to input physiological values from the most recent 24 hours of a patient’s stay in place of those measured during the first 24 hours. However, such an approach may lead to biased or inaccurate predictions, as the new patient profiles will be dissimilar to those on which the MPM was trained. Instead, one could use information from a random 24-hour interval when training MPMs, so that the profiles on which the MPM is trained align with those to which it is to be applied. In this Chapter the relative performance of these two approaches is evaluated.

**9.1 Potential Improvements Under Consideration**

Consider the situation in which there is an ICU which needs an MPM with which to make risk estimates for new patients at any time \( \geq 24 \) hours after the beginning of their ICU stay. Further suppose that one has a large ICU Database containing patient profiles from multiple ICUs, including the ICU into which the MPM is to be installed. Then the baseline approach is to apply the Standard MPMC to the entire ICU Database in order to output an MPM that is then installed into the ICU. Each time a new event is recorded for a patient that has been in the ICU for more than 24 hours, the MPM generates a new risk estimate using their updated
patient profile, with physiological events that occurred more than 24 hours earlier excluded.

Four modifications to this baseline approach are considered in this Chapter, that are intended to improve the accuracy of the output risk estimates. Details of these changes and their motivation are provided in the four proceeding Subsections.

### 9.1.1 Alternative Machine Learning Techniques

The application of different machine learning techniques is an obvious way in which one might improve the accuracy of the MPMs output by an MPMC. In this Chapter, Regularized Logistic Regression is compared with various alternatives including APACH, Random Forests and the BEAST. Nearest Neighbors, Gaussian Mixture Models and CART have been excluded because they are unlikely to offer competitive risking performance, while BART has been excluded because of issues associated with its implementation in R [26] (using the ‘BayesTree’ library), wherein crashes occur on databases containing a large ratio of features to patients. This is a known issue that has resulted in the removal of the BayesTree library from the CRAN repository.

### 9.1.2 Increased Patient Homogeneity

For both benchmarking and risking in small or new ICUs, MPMs must be trained on patients from other ICUs because the ICUs on which they are to be applied have insufficient patient records of their own on which to train them. But for risking in large established hospitals, it is viable for their MPMs to be trained using only their own patients. The potential benefit of doing this is that the patient population will be more homogeneous than for a multi-hospital database, both in terms of the patients having a more similar case-mix and in terms of the feature definitions being more consistent than they would be for a large group of hospitals. The effects on risking performance of going from a large non-homogeneous multi-ICU population to a small homogeneous single-ICU population is one focus of this Chapter.
9.1.3 More Prolific Feature Extractor Inducers

The Standard Extractor outputs only clinically important and well-defined features. However, these features represent just a small fraction of the patient information that is routinely recorded in ICUs. A potential alternative to the Standard Extractor is to retain many more features, including those that are either poorly-defined or potentially of little clinical relevance, and then leave the learner inducer to determine which features are predictive of patient mortality. This approach is embodied by the Extended Extractor Inducer defined in Section 3.4. The relative performance of the Standard and Extended Extractor Inducers is one focus of this Chapter.

9.1.4 Alternative Prediction Times

Continuous predictions are necessary for the use of MPMs in risking. One simple way to achieve this is to use an MPM trained using the Standard MPMC and then input physiological values from the most recent 24 hours of a patient’s stay in place of those measured during the first 24. However, such an approach may lead to biased or inaccurate predictions, as the new patient profiles are dissimilar to those used to train the MPM. Instead, one could use information from a random 24-hour interval when training MPMs, as is embodied in the Stochastic Prediction Time Inducer (Subsection 3.2.1). Investigating the relative performance of these two approaches is one focus of this Chapter. For both approaches, the prediction time (measured relative to ICU admission) is included as an additional feature, as it was confirmed to be predictive in past investigations [23].

9.2 Assessing Performance Improvements

The methods for altering MPM performance considered in the previous Section encompass four types of learner inducer, two types of feature extractor inducer, two patient subsets
9.2 Assessing Performance Improvements

and two prediction time inducers. It is further possible to apply different patient subsets
and different prediction time inducers to each of the training and test sets within an MPMC
independently. This then leaves 128 combinations which would ideally be applied to a single
ICU Database in order to generate 128 performance point estimates that could be directly
compared to one another. However, such an ICU Database would have to contain patients
from numerous ICUs with sufficient information per patient to apply both the Standard
Extractor and Extended Extractor Inducer. In the absence of such an ICU Database, one can
still proceed by exploiting both the APACHE Database and the MIMIC II Database, although
in a less direct manner.

The APACHE Database has been processed in such a way that it cannot have continuous
models trained on it or applied to it. It also lacks the large number of features required for
the application of the Extended Extractor Inducer. The MIMIC II Database contains only a
small number of ICUs from a single hospital, and therefore cannot be used to make patient
homogeneity comparisons. It also lacks the information required for the application of the
Standard Extractor, and therefore cannot be used to directly assess the performance of the
Standard MPMC. The fact that neither the APACHE Database nor the MIMIC II Database
can have both the Standard Feature Set and the Extended Feature Set evaluated on them is
what motivated the inclusion of the Reduced Feature Set, as a means of indirectly comparing
the two Databases.

9.2.1 Option Set Definitions

Within these constraints various ‘option sets’ are considered, with each containing one option
from each of the following groups:

1. Training patients

   (a) Patients from the APACHE Database.
(b) Patients from one of the 3 largest hospitals in the APACHE Database.

(c) Patients from the MIMIC II Database.

2. **Test patients**

   (a) Patients from the APACHE Database.

   (b) Patients from one of the 3 largest hospitals in the APACHE Database.

   (c) Patients from the MIMIC II Database.

3. **Training prediction time inducer**

   (a) 24h Fixed

   (b) Stochastic

4. **Test prediction time inducer**

   (a) 24h Fixed

   (b) Stochastic

5. **Feature extractor inducer**

   (a) Standard; \{\{APACH\}, \{RLR_1\}, \{RF_1\}, \{BEAST_1\}\}

   (b) Reduced; \{\{RLR_3\}, \{RF_3\}, \{BEAST_3\}\}

   (c) Extended; \{\{RLR_2\}, \{RF_2\}, \{BEAST_2\}\}

The application of the Fixed Prediction Time Inducer to patients in the training set and the Stochastic Prediction Time Inducer to patients in the test set and vice versa is viable because the Fixed and Stochastic Prediction Time Inducers exclude patients according to the same criteria: an ICU length of stay of less than 24 hours. This mixture of Prediction Time Inducers simply implies that the patient profiles in any training set retain events from a
24-hour interval that begins at ICU admission, while the patient profiles in any test set retain events from a random 24-hour interval.

The option sets that include either training or test patients ‘from one of the 3 largest hospitals in the APACHE Database’ have been included to try and bridge the gap between the multi-hospital APACHE Database and the single-hospital MIMIC II Database. After stay exclusions the MIMIC II Database retains a total of 13,224 patients, while the APACHE Database contains three hospitals that each contain at least 13,224 patients. To facilitate comparisons, option sets incorporating 2b therefore apply their associated MPMC a total of three times, with each application using only 13,224 patients from one of the three largest ICUs in the APACHE Database for MPM training and hyper-parameter selection.

If an option set includes options 1a and 2b, then all patients from the APACHE Database are used for training but only patients from the ICUs within the 3 largest\(^1\) hospitals in the APACHE Database are used for testing. For such option sets, the majority of MPMC application steps proceed as usual, except that after the outer risk array is generated it must have all entries removed that correspond to patients from outside of the three largest hospitals.

For option sets that incorporate both options 1b and 2b the application of the MPMC is adjusted as follows:

1. The data extraction, stay exclusions and outer partitioning proceed as usual.

2. For \(\alpha \in \{1, 2, 3\}\):
   
   (a) \(\beta\) is set to be a copy of the outer partitioning matrix.

   (b) All entries in \(\beta\) corresponding to patients outside of the \(\alpha\)’th largest hospital are set to zero.

   (c) A total of 13,224 rows (patients) in \(\beta\) are marked.

\(^1\)The ‘largest’ hospitals are taken to be those with the greatest number of admissions in the APACHE Database.
(d) Hyper-parameter selection and risk array generation proceed using $\beta$ with only the marked rows used in hyper-parameter selection and MPM training.

3. The three risk arrays are then combined and performance evaluation and uncertainty evaluation performed as usual.

### 9.2.2 Option Set Selection

A total of ten option sets are considered in this Chapter, as listed in Table 9.1. Option Set 1 applies the Standard Feature Extractor Inducer with the APACHE Database and the 24h Fixed Prediction Time Inducer to both training and test patients. Out of all possible option sets, this is therefore the one which most closely resembles the Standard MPMC. This similarity then makes Option Set 1 a natural starting point for the performance investigations of this Chapter.

Ideally the next option set to consider would be the same as Option Set 1, except with the prediction time inducer used on the testing patients changed from 24h Fixed to Stochastic. This would then provide the ideal baseline for the internal risking problem considered in the previous Section. However, this is not a viable option set because the APACHE Database does not contain the information required to have the Stochastic Prediction Time Inducer applied to it. Instead this transition to the Stochastic Prediction Time Inducer for testing patients must be delayed until further option sets have formed a bridge from Option Set 1 to an option set incorporating the MIMIC II Database.

Option Set 2 is the same as Option Set 1, except that it replaces the Standard Feature Extractor Inducer with the Reduced Feature Extractor Inducer. Option Set 3 is the same as Option Set 1 except that MPMs are only tested on patients from one of the three largest hospitals as opposed to patients from all hospitals. These two transitions are necessary to move from the Standard MPMC towards an MPMC that can be applied to the MIMIC II Database, because the MIMIC II Database contains neither multiple hospitals nor the
<table>
<thead>
<tr>
<th>Option Set</th>
<th>Patients</th>
<th>Prediction Time Inducer</th>
<th>Feature Extractor Inducer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
<td>Training</td>
</tr>
<tr>
<td>1</td>
<td>1a) All APACHE ICUs</td>
<td>2a) All APACHE ICUs</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>2</td>
<td>1a) All APACHE ICUs</td>
<td>2a) All APACHE ICUs</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>3</td>
<td>1a) All APACHE ICUs</td>
<td>2b) One APACHE ICU</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>4</td>
<td>1a) All APACHE ICUs</td>
<td>2b) One APACHE ICU</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>5</td>
<td>1b) One APACHE ICU</td>
<td>2b) One APACHE ICU</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>6</td>
<td>1b) One APACHE ICU</td>
<td>2b) One APACHE ICU</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>7</td>
<td>1c) MPMIC II Database</td>
<td>2c) MPMIC II Database</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>8</td>
<td>1c) MPMIC II Database</td>
<td>2c) MPMIC II Database</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>9</td>
<td>1c) MPMIC II Database</td>
<td>2c) MPMIC II Database</td>
<td>3a) 24h Fixed</td>
</tr>
<tr>
<td>10</td>
<td>1c) MPMIC II Database</td>
<td>2c) MPMIC II Database</td>
<td>3b) Stochastic</td>
</tr>
</tbody>
</table>

Table 9.1 Table specifying which options are included in each option set considered in this Chapter. Options 1-5 specify the patients in the training set, the patients in the test set, the training set prediction time inducer, the test set prediction time inducer and feature extractor inducer used, respectively, as defined in Subsection 9.2.1.
9.2 Assessing Performance Improvements

information necessary for the application of the Standard Feature Extractor Inducer. Option Set 4 is the same as Option Set 1 except that both the above transitions have been applied.

Option Set 5 is the same as Option Set 3, except that the MPMs are trained using only patients from one of the three largest hospitals rather than the patients from all hospitals. Moving from Option Set 3 to Option Set 5 then provides an example of the transition from a large non-homogeneous multi-hospital population to a small homogeneous single-hospital population, as required for the patient homogeneity investigation considered in Subsection 9.1.2. The same is true on moving from Option Set 4 to Option Set 6, except that these use the Reduced Feature Extractor Inducer in place of the Standard Feature Extractor Inducer.

Option Set 7 is the same as Option Set 6, except that the MPMs are trained and tested on the MIMIC II Database instead of on the patients from the largest hospitals in the APACHE Database. Moving from Option Set 6 to Option Set 7 represents the smoothest transition from the APACHE Database to the MIMIC II Database. Option Set 8 is the same as Option Set 7, except that the Reduced Feature Extractor Inducer is replaced with the Extended Feature Extractor Inducer. The increase in performance on moving from Option Set 7 to Option Set 8 minus the increase in performance on moving from Option Set 6 to Option Set 5 should provide an indication of the increase in performance that one might expect if the Standard Feature Extractor Inducer was replaced with the Extended Feature Extractor Inducer. This should then help resolve the Feature Extractor Inducer investigation considered in Subsection 9.1.3.

Option Set 9 is the same as Option Set 8, except that the prediction time inducer used on the test patients has been changed from 24h Fixed to Stochastic. Option Set 9 then represents the first example of an MPMC that can make continuous predictions. Option Set 10 is the same as Option Set 9, except that the prediction time inducer used on the training patients has been changed from 24h Fixed to Stochastic. The difference in performance between Option
Table 9.2 Specification for a set of MPMCs used to assess different option sets. The Prediction Time Inducer and $MPMI^2$ is not defined, with these options depending on the option set with which the MPMC is applied. Descriptions of the various steps are linked through Table 5.4.

Set 9 and Option Set 10 should then resolve the prediction time investigation considered in Subsection 9.1.4.

### 9.3 Evaluation Methods

For $\pi \in \{1, 2, \ldots, 10\}$ an MPMC is created by combining the $\pi$’th option set with the set of MPMCs defined in Table 9.2. This is then applied in order to generate an AUC performance point estimate and an AUC standard error estimate.

The use of 10-Repeat 2-Fold Patient CV (9.3), Mean Second (12.2) and Set Comparison (13.1) is in line with the findings of Chapter 8, wherein this combination was found to be the most appropriate party for assessing internal risking performance. The AUC has been used in place of the SLL because most readers will be more familiar with the former and will find interpreting differences in AUCs easier than interpreting differences in SLLs.
9.4 Results and Discussion

The performance of the MPMs trained using different option sets are presented in Figures 9.1 and 9.2. Each number in these plots represents a combination of an option set and a learner inducer. The value of each number dictates its corresponding option set as specified in Table 9.1. The x-coordinate of each number dictates its corresponding AUC performance point estimate, with larger values indicating superior performance. Each pair of error bars is centered on the performance point estimate, with a half-width of 1.96 times the corresponding standard error estimate. In Figure 9.1, the y-coordinate of each number dictates its corresponding learner inducer. In Figure 9.2, all numbers correspond to the BEAST irrespective of their y-coordinate. Beyond the AUCs displayed in these figures, the Standardized Mortality Ratios for the BEAST applied on Option Sets 9 and 10 were 1.049 and 1.005, respectively.

The top entry in Figure 9.1 is for the APACH learner inducer and its AUC of 0.847 is substantially less than that of 0.88 stated in the original APACHE IV paper [87]. This difference can be explained in terms of the different prediction time inducers associated with these two performance point estimates. The value of 0.88 is associated with the Deterministic Variable Prediction Time Inducer which includes patients with an ICU length of stay of greater than 4 hours, while the AUC of 0.847 (as is the case with every AUC presented in Figures 9.1 and 9.2) is associated with the 24h Fixed Prediction Time Inducer, with only patients with a length of stay of greater than 24 hours included. The response for the patients with a length of stay of between 4 and 24 hours is potentially much easier to predict than that for other patients, because at prediction time they are within 4 hours of either death or discharge and hence likely to be either very ill or relatively well. As such, removing these patients naturally reduces discrimination by removing patients whose outcomes are particularly easy to predict.
Fig. 9.1 Performance plots providing a comparison of the risking performance of different learner inducers. Each number represents a combination of an option set and a learner inducer. The value of each number corresponds to the index of the option set as defined in Table 9.1 and the y-coordinate specifies the learner inducer which it applied. The x-coordinate of each number corresponds to the performance point estimate output by the given combination of option set and learner inducer. Such performance point estimates are measured using the Area Under the Curve (AUC), with larger values indicating superior performance. Each pair of error bars is centered on the performance point estimate and has a half-width of 1.96 times the standard error estimate for the given combination of option set and learner inducer. Red points correspond to option sets incorporating the reduced features, blue points correspond to the standard features and green points correspond to the extended features.

### 9.4.1 Machine Learning Techniques

The performance of the different learner inducers is illustrated in Figure 9.1. Regularized Logistic Regression (RLR) offers a statistically significant but insubstantial improvement in performance over APACH, with this difference potentially due to the regularization and complex preprocessing steps applied for RLR. Random Forests (RF) outperform RLR when applied to option sets with few features (red and blue), potentially because they offer superior model flexibility that allows them to make better use of the large number of patients in the APACHE Database. However, the performance of RLR is superior to that of Random Forests on the Extended Feature Set (green), potentially because RLR’s high degree of regularization protects it from over-fitting. The BEAST solidly outperforms the other machine learning
9.4 Results and Discussion

Fig. 9.2 Performance plots providing a comparison of the risking performance of the different option sets defined in Table 9.1. The x-coordinate of each number corresponds to the performance point estimate output by the application of the BEAST with the corresponding option set. Such performance point estimates are measured using the Area Under the Curve (AUC), with larger values indicating superior performance. Each pair of error bars is centered on the performance point estimate and has a half-width of 1.96 times the standard error estimate for the given combination of option set and learner inducer. The y-coordinate of each number corresponds to the patients in the test set. Red points correspond to option sets incorporating the reduced features, blue points correspond to the standard features and green points correspond to the extended features.

techniques considered in this Chapter in both the many-patient (red and blue) and the many-feature (green) regimes, with a moderate increase in AUC relative to APACH.

9.4.2 Patient Homogeneity

The relative performance of MPMs trained on a small homogeneous ICU Database versus those trained on a large non-homogeneous ICU Database is provided in the top two rows of Figure 9.2. The MPMs associated with Option Sets 3-6 are each evaluated on the same patients, with Option Sets 3 and 4 trained using all patients and Option Sets 5 and 6 trained on only the patients from the same ICU as the patients on which the output MPM is applied. The significantly larger AUC for Option Sets 3 and 4 versus Option Sets 5 and 6 indicates that, provided features are well-defined across ICUs, large non-homogeneous ICU Databases such as the APACHE Database outperform small homogeneous ones.

That the difference in the AUC between Option Sets 3 and 5 is greater than that between Option Sets 4 and 6 indicates that the relative benefit of large non-homogeneous
ICU Databases may increase as the number of available features increases, in line with expectations. This suggests that if a consistent schema could be developed and the MIMIC II Database extended to include patients from many more ICUs, then one should be able to produce MPMs that offer greatly improved risking performance.

### 9.4.3 Feature Extractor Inducers

The relative performance of MPMs trained using different Feature Extractor Inducers is provided in Figure 9.2. The performance point estimate for Option Set 5 minus that for Option Set 6 represents the increase in performance on moving from the Reduced Feature Extractor Inducer to the Standard Extractor Inducer when applied to and evaluated on a single large ICU from the APACHE Database. The associated increase in AUC of 0.013 is statistically significant and quite substantial. The performance point estimate for Option Set 8 minus that for Option Set 7 represents the increase in performance on moving from the Reduced Feature Extractor Inducer to the Extended Feature Extractor Inducer when applied to and evaluated on the MIMIC II Database. The associated increase in AUC of 0.035 is statistically significant and substantial. Although differences in AUCs are not additive, one might still expect to see a substantial increase in AUC on moving from the Standard Feature Extractor Inducer to the Extended Feature Extractor Inducer.

That the difference in the AUC between Option Sets 3 and 4 is greater than the difference between Option Sets 5 and 6 indicates that the relative benefit of feature sets with additional features is even greater for large non-homogeneous ICU Databases than for small homogeneous ICU Databases, in line with expectations. This then suggests that if features in the Extended Feature Set could be consistently defined across ICUs, then increasing the number of features recoded in APACHE-style ICU Databases could greatly increase the risking performance of output MPMs.
9.4.4 Prediction Times

The effects of different Prediction Time Inducers are illustrated in the bottom row of Figure 9.2. Moving from Option Set 8 to Option Set 9 corresponds to a change from making predictions using information from the first 24 hours of a patient’s stay to using information from a random 24-hour interval during the patient’s stay. The vast increase in performance associated with this transition is due to the use of more up to date information.

Moving from Option Set 9 to Option Set 10 represents the transition from training MPMs using information from the first 24 hours of a patient’s stay to using information from a random 24-hour interval during the patient’s stay. The same MPMI is applied to the same patients in both cases, but the different patient profiles in the training sets give rise to different MPMs. The two MPMs are then evaluated on the same patients using the same patient profiles. The modest increase in performance associated with the transition from Option Set 9 to Option Set 10 suggests that ensuring that the patient profiles in the training set are from the same distribution as the patient profiles in the test set has little effect on the degree of discrimination offered by such MPMs. However, the SMR for Option Set 9 of 1.049 is considerably worse than that of 1.005 for Option Set 10. This likely relates to lead-time bias, with patients further into their ICU stay having had some of their abnormal physiology suppressed by treatments such as anti-hypertensives. The very poor calibration of Option Set 9 relative to Option Set 10 suggests that the latter approach should be favored.

9.5 Conclusions

Altering the Standard MPMC to incorporate a more complex inducer such as the BEAST should offer a substantial increase in risking performance. Such a change would not require the gathering of any additional information, as a variety of methods already exist that can be applied directly to existing ICU Databases. The implementation of complex learner inducers
9.5 Conclusions

would preclude the manual evaluation of patient risk, as is possible for simple MPMs such as the APACHE IV Model. However, the rapid uptake of electronic medical records within ICUs [15] lessens this issue.

MPMs trained using a multi-hospital ICU Database were shown to offer a moderate increase in risking performance over those trained using a single hospital ICU Database, with evidence that the magnitude of such improvement increases along with greater learner complexity. It was also shown that applying a more prolific feature extractor can substantially increase the risking performance of the MPMs output by an MPMC. These trends suggest that creating a large multi-hospital ICU Database, similar to the APACHE Database but containing a very large amount of per patient information similar to that within the MIMIC II Database, should provide the opportunity to construct and apply extremely accurate risking MPMs. However, such an ICU Database would require a vast amount of work as, although such information is already routinely recorded in many ICUs, this additional information is rarely stored in an easily accessible format and features would need to be defined very carefully such that their meaning is consistent across institutions. Still, the benefits of creating such a large and rich ICU Database would be far from limited to its use in constructing and applying MPMs. Further opportunists would likely arise in settings such as treatment decision support systems, with the aim of identifying which patients are suffering from specific diseases.

Even in the absence of such new ICU Databases, the MPMs trained using Option Set 10 should offer a substantial improvement over the MPMs output by the Standard MPMC with regards to internal risking. If such MPMs did provide more accurate risk predictions than those of nurses, then it might be able to form a viable deterioration alarm. However, before such a method could be used for treatment decision support, a set of further studies would need to be carried out including:

1. A validation study to ensure the stated improvements hold in other ICU Databases.
2. A study comparing the accuracy of predictions output by such MPMs to those of nurses and MPMs trained using state of the art alternative methods.

3. An impact study to ensure that a deterioration alarm based on such an MPMC improves ICU patient outcomes.

It is also important to note that the generalization error estimates considered throughout this thesis have ignored model drift, wherein patient populations and treatments change over time such that MPMs become less accurate, particularly in terms of their calibration, as they are applied to patients whose admission dates fall further outside of the range of admission dates of the patients on which the MPMs were trained.
Chapter 10

Conclusions and Further Work

Introduction

This thesis has examined the development of mortality prediction models (MPMs) for use in intensive care. Mortality prediction model constructors (MPMCs) were introduced as methods for converting an ICU Database into either a set of ICU performance evaluations (one for each ICU in the ICU Database to which the MPMC was applied) or a set of one or more MPMs with associated performance measures.

The Acute Physiology and Chronic Health Evaluation (APACHE) Database and version 2.6 of the Multi-parameter Intelligent Monitoring for Intensive Care (MIMIC II) Database were introduced as two substantial ICU Databases. The APACHE Database was characterized as containing a moderate amount of information for each of 231,019 ICU admissions spread across 109 ICUs, while the MIMIC II Database contains a large amount of information for each of approximately 30,000 ICU admissions within a single hospital.

The main MPMCs of the MPM literature were discussed and their common steps identified and combined to form the ‘Standard MPMC’. A variety of alternatives to each step in the Standard MPMC were motivated and detailed. The remainder of the thesis then used
the APACHE and MIMIC II Databases to compare each Standard MPMC step against its alternatives, in order to identify the most appropriate sets of steps for different use cases.

**Simulations**

A mortality prediction model inducer (MPMI) was introduced as a method for creating an MPM. Methods were presented for simulating both ICU Databases and MPMIs, with which to support investigations into MPM performance evaluation and partitioning. These simulations allowed ICUs to vary in terms of both the quality of care that they provide and the case-mix of their patients, while MPMs were allowed to vary in terms of both their Specific Ability (the extent to which they could account for intra-ICU mortality rate variability) and their Case-mix Ability (the extent to which they could account for inter-ICU mortality rate variability).

The ‘Representative MPMC’ was created to include a wide range of MPMIs of the kind that one might expect to see used in practice. A set of ‘risk array statistics’ were defined with which to quantify the key properties associated with an MPMC’s outputs. Sets of simulation parameters were found for which the outputs from the application of the risk array statistics were the same whether they were applied to the outputs of such simulations or to the outputs of the application of the Representative MPMC to the APACHE Database. These sets of simulation parameters indicated that the correlation between the Specific Ability and Case-mix Ability of MPMs is of moderate strength, with a Kendall correlation of approximately 40%. It was further inferred that even the best MPMs are far from able to fully account for differences in ICU case-mix.

**Performance Evaluators**

It was demonstrated that the standard approach to creating ICU quality of care p-values only holds if MPMs are able to fully account for differences in ICU case-mix. Given that
even the best MPMs are far from able to fully account for differences in ICU case-mix, this then indicates that such ICU quality of care p-values are flawed. To overcome this issue the ‘Posterior Model’ was introduced as an alternative method for generating ICU quality of care p-values, in which the inability of MPMs to fully account for differences in ICU case-mix is integral.

Both the risking and benchmarking performance of an MPM evaluator were taken to be measures of how often the evaluator correctly selected, out of two possible MPMs, the one that was best for either risking or benchmarking, respectively. Simulations were used to demonstrate that the Area Under the Curve (AUC) is an excellent measure of MPM risking performance (99% MPM discrimination) but a relatively poor measure of MPM benchmarking performance (62%), when applied to an ICU Database containing a large number (109) of ICUs. In contrast a chi-squared test ($\chi^2$) with patients grouped by ICU was found to be a poor measure of MPM risking performance (63%) but a good measure of MPM benchmarking performance (86%), while the Hosmer-Lemshow Statistic (H-L) was found to be a poor measure of both MPM risking (60%) and MPM benchmarking (53%) performance. These findings suggest that switching from the AUC and H-L to some alternative performance evaluator such as the $\chi^2$ could offer substantially improved benchmarking MPM performance comparisons, when applied to ICU Databases that contain a large number of ICUs.

**Partitioning**

Simulations were used to compare partitioning approaches that randomly split patients without regard to ICU membership, and approaches that randomly split ICUs. MPMCs that randomly partitioned patients were found to output highly over-optimistic estimates of MPM benchmarking performance, while the estimates output by MPMCs that partitioned ICUs were found to exhibit minimal bias. In contrast, these two types of partitioning approach were found to output very similar risking performance estimates. A variety of methods for the
creation of standard error estimates with which to supplement MPM performance estimates were defined and tested, with most dismissed as overly biased and some verified as providing useful outputs.

**Accuracy**

MPMCs incorporating Regularized Logistic Regression, Random Forests and the Bayesian Ensemble of Additive Sigmoidal Trees (BEAST) were applied to a variety of ICU Datasets in order to evaluate the relative effects on risking performance of different machine learning techniques, types of ICU Database, feature subsets and prediction times. The BEAST was found to be the best performing machine learning technique on all datasets, offering an increase in AUC from 0.845 to 0.860 compared to a replica of the approach used to train the APACHE IV Model, when applied to the APACHE Database.

When tested on patients from a single large hospital, it was found that MPMs trained using only patients from ICUs within the same hospital as the patients on which it was tested (AUC=0.868) tended to be inferior to MPMs trained using patients from all ICUs within the same ICU Database (AUC=0.876); this indicates that MPMs trained on large non-homogeneous ICU Databases tend to outperform those trained on small homogeneous ICU Databases, provided features can be defined consistently across ICUs.

Altering patient profiles such that they contain much more clinical information per patient was shown to offer a substantial increase in risking performance (AUC=0.876 to AUC=0.890) for MPMCs applied to a small homogeneous ICU Database. For risking, MPMs trained using stochastic prediction times were shown to outperform those trained using a fixed prediction time both in terms of calibration (SMR=1.005 versus SMR=1.05) and discrimination (AUC=0.930 versus AUC=0.925).
Translation into Clinical Practice

The results of this thesis have implications for how predictive models should be developed for use in critical care units. One of the main problems in comparing ICUs is the highly variable case-mix among them. Differences in diagnoses and severity of illness, if not accounted for properly, can result in inadequate benchmarking results. A common mistake that affects many critical care MPMCs is the way in which they partition admissions into development and validation data sets, wherein splitting is performed randomly across patients, while this thesis shows that it is preferable to partition on an ICU-basis. Further, the inclusion of a benchmarking performance evaluator such as the $\chi^2$ Statistic helps in choosing among different models. It is therefore recommended that these modifications are adopted by researchers and institutions that create, compare and distribute benchmarking MPMs.

Benchmarking MPMCs aim to facilitate ICU performance evaluations through the creation of ICU quality of care estimates. The investigations of this thesis indicate that the standard approach of creating and interpreting such quality of care estimates as p-values can result in a misleading comparison of ICUs. Instead, it is recommended that ICU quality of care estimates are generated using a method, such as the Posterior Model, that is able to account for the limited degree to which MPMs are able to adjust for differences in ICU case-mix. Such an approach would requires developers of MPMs to assess the degree to which their models are able to account of differences in ICU case-mix. This would lead to more readily interpretable ICU quality of care estimates and more appropriate p-values.

This thesis also demonstrated that the choice of ICU database affects predictions. The choice of ICU database type should adhere to the following order of preference (from lowest to highest) conditioning on data availability: A single ICU cohort containing a moderate number of features; a multi-ICU cohort containing a moderate number of consistent features (such as the APACHE Database); a single ICU cohort containing a large number of features (such as the MIMIC II Database); and a multi-ICU cohort containing a large number of
consistent features. Further, both the use of more versatile machine learning techniques, such as the BEAST, and the transition from a fixed prediction time inducer to a stochastic prediction time inducer, should help provide more accurate predictions. This might lead to the construction of more accurate MPMs for generating alerts of deterioration in an individual patient’s acuity of illness.

Limitations and Further Work

When choosing the content to include in a thesis, one must consider various constraints including available time, expertise, data and the agreed topic of research. One must make a trade-off between depth and breadth, with the present work leaning slightly towards the latter. This choice was made because of the substantial amount of work required to create a viable set of simulations and MPMC framework, that then lent themselves to answering a range of research questions once developed. In spite of this, the scope of this thesis does not include compartmentalization or corruption, which are among the key issues preventing the rapid clinical uptake of ICU MPMs, and which are only touched on superficially because of the author’s expertise.

There are several clear ways in which the depth of this thesis could be increased, for example by considering further machine learning techniques such as support vector machines and neural networks, by greatly increasing the analysis of the BEAST and it’s predictions, or by including additional partitioning approaches such as those that balance the number of survivors and non-survivors within training and test sets. However, such extensions would not contribute greatly to the thrust of the present work. Instead, two potential avenues of further work that could greatly improve the depth of this thesis are described in the proceeding Subsections, alongside the limitations that they might potentially overcome.
Simulation Parameter Confidence Intervals

Ideally, the simulation parameters derived in Chapter 6 would have been presented with appropriate confidence intervals. This would have then enabled quantification of the uncertainties associated with the results of Chapters 7 and 8, as these depends on the uncertainty in the simulations parameters used in their derivation.

One could produce uncertainty estimates by repeatedly regenerating the simulation parameters while bootstrapping over the patients and $MPM_1^1$s that are included in the Representative MPMC. However, such a step in isolation would provide a misleading set of confidence intervals because of the nature of the $MPM_1^1$s that are included in the Representative MPMC. These $MPM_1^1$s are not a random sample from some large and well defined population. Instead, they are a set of $MPM_1^1$s chosen by the author in the hope that their properties will reflect those of the population of $MPM_1^1$s that one might expect to see implemented in practice. Bootstrapping the $MPM_1^1$s in the Standard MPMC would therefore not provide meaningful confidence intervals, because such an approach would not account for the bias introduced by differences in the $MPM_1^1$s in the Representative MPMC relative to the population of MPMCs that they intend to represent.

One could attempt to mitigate this issue by replacing the $MPM_1^1$s included in the Standard MPMC with the $MPM_2^2$ composed of an unbiased sample from all of the $MPM_1^1$s that have been implemented in the MPM literature. Such an approach would be extremely time consuming, but is perhaps the only way to generate meaningful sets of simulation parameter confidence intervals. However, such an approach would remain biased as the calculated simulation parameters and their associated confidence intervals would be for $MPM_1^1$s that have been used in the past rather than for the $MPM_1^1$s that one might expect to be used in the future. Instead, one might look to go beyond the MPM literature and instead search for a more advanced set of binary machine learning techniques and preprocessor inducers that
could be combined with the feature inducers of the MPM literature, in order to generate a very broad set of $MPMI^1$s to include in the Representative MPMC.

Including a better mix of $MPMI^1$s in the Representative MPMC would also allow for an important addition to the simulations of Chapter 6. Applying the same $MPMI^0$ to different training sets outputs different MPMs. The extent to which these MPMs differ from one another is not incorporated into the simulation parameters. Instead, the learner applied during simulations effectively has a degree of regularization that is set by the author in the hope that it will provide a similar degree of MPM variability, measured across multiple applications of the same $MPMI^0$ to different training sets, as one might expect to find in the $MPMI^0$s in the representative MPMC. As an alternative, one could introduce two additional simulations parameters: one that defines the mean degree of regularization for the $MPMI^0$s applied in the simulations, and one that defines the standard deviation associated with this degree of regularization. The values for these two additional simulation parameters could then be found by introducing two additional risk array statistics: one that measures the correlation between the predictions formed on the same patients by MPMs output by the same $MPMI^1$ trained on different patients, and one that measures the variability in this measure, assessed across different $MPMI^1$s. Given the state of the existing Representative MPMC, the values associated with these additional simulation parameters would be extremely difficult to measure in a reliable manner. However, altering the Representative MPMC as described previously should be sufficient to allow the values of such additional simulation parameters to be estimated with far less bias. This could then give rise to simulations that are more representative of real-world ICU Databases, and that could be used to increase the reliability of the results of Chapters 7 and 8.
Further ICU Databases

The results of this thesis are based on the application of MPMCs to two ICU Databases, the APACHE Database and the MIMIC II Database, while the fitting of simulation parameters in Chapter 6, and subsequently all results presented in Chapters 7 and 8, are based solely on the APACHE Database. Two ICU Databases similar to the APACHE Database were introduced in Chapter 2 (the ICNARC Database and the ANZICS Adult Patient Database), and using such alternatives to regenerate the results of Chapters 6-8 and ensure that they are in line with those derived for the APACHE Database would be of considerable benefit, with access as potentially the main hurdle. Further, the MIMIC II Database has now been superseded by the MIMIC III Database [42], with which select results from Chapter 9 could be updated.

In addition to the APACHE-style databases, there is at least one ICU Database that contains many more patients: the Philips eICU archives [55] is an ICU Database containing patient records for an estimated 1.5 million ICU stays. Chapter 8 included investigations into different partitioning approaches and generalization error estimation approaches for internal risking, that did not require the use of simulations. The equivalent investigation for external risking was not included because the APACHE Database did not contain a sufficiently large number of ICUs for such investigations to yield reliable results. However, the greater number of ICUs included in the Philips eICU archives might allow for such investigations to be applied, and the outcomes used to either support or discredit the findings from the equivalent simulation-biased investigations of Chapter 8.

To calculate meaningful ICU quality of care p-values, it was found that knowledge is needed of the ratio of the inter-ICU mortality rate variability attributable to each of differences in ICU quality of care and differences in ICU case-mix. However, this ratio could not be evaluated using risk array statistics because risk arrays are not affected by its value. One approach that could be used to estimate the ratio is to first create an ICU Database in which a portion of patients are assigned to ICUs at random. Differences in inter-ICU mortality rate
variability within this randomly assigned population would then be due entirely to differences in ICU quality of care, while differences within the non-randomly assigned population would be due to a combination of both differences in ICU quality of care and in ICU cases-mix. The relative sizes of the inter-ICU mortality rate variability within each population could then be used to calculate the required ratio. However, generating such an ICU Database would require a large randomized controlled trial that would be extremely difficult to perform because of geographic constraints. Instead, one might consider the less accurate alternative of consulting physicians and analyzing studies of admissions for which avoidable patient deaths are known to have occurred in order to refine our understanding of this ratio, although the amount of human effort involved in such an analysis would still be substantial.
References


[64] Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.


Appendix A

Glossary of Terms

APACHE (Acute Physiology and Chronic Health Evaluation); the APACHE Model is an ICU MPM that was trained on the APACHE Database. The APACHE Database is a large multi-ICU Database and the primary ICU Database of this thesis.

Area Under the Curve; the standard measure of MPM performance with a value of 0.5 indicating no discrimination and 1.0 indicating perfect discrimination.

Benchmarking; the act of assessing the quality of care provided by an ICU through the application of an MPM. A benchmarking MPM is an MPM that is designed for use in benchmarking. The benchmarking performance of an MPM is a measure of how effectively the MPM can be used to measure the quality of care provided by ICUs.

Broad $MPM_1^1$ Subset; an instance of an $MPM_2$ containing a wide range of $MPM_1^1$s.

Candidate Stays; all ICU stays within a given set of ICUs during a specific interval.

Care Component; the increase in risk for an average patient, treated in a given ICU, over the same patient had they been treated in an average ICU.

Case-mix; an ICU’s case-mix specifies the distribution of health states from which the ICU’s patients are drawn. An ICU’s case-mix component specifies the extent to which an average patient from that ICU is more unwell than an average patient from an average ICU. An MPM’s case-mix ability specifies the extent to which it can account for an ICU’s case-mix
component.

**Event**: a single patient record, such as a breathing rate measurement or the administration of a drug.

**External MPM**: an MPM that is intended for use on patients from ICUs not included in the ICU Database on which it was trained.

**External Partitioning**: splitting patients into validation and non-validation sets.

**Generalization error**: a measure of how well an MPM will perform when applied to new patients.

**Hosmer-Lemshow Statistic**: a measure of an MPM’s performance.

**Hospital Floor**: any non-ICU hospital department.

**Hospital Stay**: an uninterrupted interval during which a given patient remains in a single hospital.

**ICD9 Codes**: a system for classifying diseases maintained by the World Health Organization.

**ICU (Intensive Care Unit)**: a specialized hospital department that provides life support to critically ill patients.

**ICU Database**: a large collection of ICU patient medical records and the object to which an MPMC is applied in order to output an MPM.

**ICU Evaluator**: a method for assessing the performance of ICUs.

**ICU Stay**: an uninterrupted interval during which a given patient remains in a single ICU.

**object* Inducer**: a method for creating an *object*.

**Internal MPM**: an MPM that is intended for use solely on patients from the same set of ICUs as those in the ICU Database on which it was trained.

**Internal Partitioning**: splitting patients from a non-validation set into training and testing sets.

**Learner**: a function that maps a processed design matrix onto a risk estimate.

**Membership**: an integer specifying the ICU to which a patient is assigned.
MIMIC (Multiparameter Intelligent Monitoring in Intensive Care) II Database; a large multi-ICU Database from a single hospital containing information-rich patient profiles, and the secondary ICU Database of this thesis.

MPM (Mortality Prediction Model); a function that maps a patient profile onto a risk estimate.

MPMC (Mortality Prediction Model Constructor); a function that takes as inputs an ICU Database and that outputs (among other things) an MPM or two or more MPM performance estimates or a set of ICU quality of care estimates.

MPM\(_0\) (Mortality Prediction Model Inducer); a method for converting a set of patient profiles and a response vector into an MPM.

MPM\(_1\); a set of one or more MPM\(_0\)s that all belong to the same class of machine learning technique but differ in terms of their associated hyper-parameter values.

MPM\(_2\); the complete set of one or more MPM\(_1\)s that are applied by an MPMC.

MPM Evaluator; a method for assessing the performance of an MPM.

Null Component; the risk for an average patient treated in an average ICU.

Null Model; an MPM trained without any features.

Patient Profile; all of the clinically relevant information known about a given patient before prediction time, excluding information directly related to a patient’s membership.

Partitioning Matrix; an object used to specify how patients are split into training and testing sets in order to avoid over-fitting when evaluating the performance of either MPM\(_0\)s or MPM\(_1\)s.

Performance MPM\(_1\) Subset; an instance of an MPM\(_2\) containing a small number of MPM\(_1\)s that are expected to generate highly accurate MPMs.

Performance Point Estimate; a measure of how well an MPM is expected to perform when applied to new patients.

Prediction time; the time at which a prediction is to be made for a given patient.
**Primary ICD9 Code:** an identifier for the disease thought to be the primary cause of a patient’s death.

**Processed Database:** the combination of a profile, a response and a membership, for each patient within a large retrospective cohort.

**Quality of Care:** the extent to which an ICU increases a patient’s probability of survival relative to treatment in an average ICU.

**Random ability:** the extent to which an MPM can account for an ICU’s case-mix component.

**Random component:** the risk for a given patient relative to an average patient both from, and treated in, the same ICU.

**Real-time:** a type of MPM that can be used over a range of prediction times for a single patient.

**Representative MPMC:** an instance of an MPMC that trains a large number of MPMs and that is used to fit simulation parameter values.

**Response:** an indicator for whether or not a given patient went on to develop a particular state of health within a specified time; usually death before hospital discharge.

**Risk:** the probability of a patient developing a particular response.

**Risk Array:** an array containing the predictions generated by the MPMs produced during the application of a single MPMC.

**Risk Array Statistics:** a set of statistics designed to capture the key attributes associated with a given instance of a risk array.

**Risking:** an assessment that informs the care of an individual patient through the application of an MPM, otherwise known as ‘patient specific decision support’. A **risking MPM** is an MPM that is designed for use in risking. The **risking performance** of an MPM is a measure of how effectively the MPM can be used in risking.

**Specific Illness:** The extent to which a patient is more ill than an average patient admitted to the same ICU.
**Standard Error Estimate**; an estimate for the standard error associated with a given performance point estimate.

**Standardized Log Likelihood (SLL)**; a measure of an MPM’s performance.

**Standard MPMC**; an instance of an MPMC that combines the common elements of the MPMs within the MPM literature.

**State**; the sum of all clinically relevant information for a given patient at prediction time, excluding information directly related to the patient’s membership.
Appendix B

Glossary of Symbols

$\chi^2$; the $\chi^2$ Statistic, an MPM performance evaluator (defined in Subsection 5.4.2).

$a$; an index over partitions.

$A$; the total number of partitions applied by a given partitioning matrix.

$argmax$; a function that outputs the set of arguments that maximize the input function.

$b$; an index over simulation blocks.

$B$; the total number of simulation blocks associated with a given simulation.

$c_k$; the Case-mix Component associated with the $k$’th ICU. It is a measure of the extent to which the patients in the $k$’th ICU are more unwell than those in an average ICU.

$\hat{c}_i$; an estimate for the Case-mix Component for the ICU in which the $i$’th patient was treated.

$C_m$; the Case-mix Ability of the $m$’th MPM. It is a measure of the extent to which the given MPM can account for ICU case-mix.

$e$; an event. Each event is specified as a combination of five values: $e^1$ (the type of event), $e^2$ (the time-stamp for when the event occurred), $e^3$ (the time-stamp for when the event was recorded), $e^4$ (the value recorded for the event), and $e^5$ (the index of the patient to whom the event relates).

$E$; the expectation. For an arbitrary array, $\beta$, and a set of one or more indices, $\alpha$, $E_\alpha(\beta)$ denotes the expectation of $\beta$ across all possible values of $\alpha$; for example $E_i(y_i) = \frac{1}{t} \sum_{i=1}^{t} (y_i)$. 
$f$; an index over the test sets associated with a given partition.

$F$; the number of test sets associated with a single partition from a given partitioning matrix.

$G^k$; the proportion of times that a given MPM evaluator fails to identify the best MPM for a given task, evaluated using simulation parameter set $k$.

$h_i$; the index of the ICU in which the $i$'th patient under consideration was treated.

$i$; an index over patients or ICU stays.

$I$; the total number of patients or ICU stays under consideration.

$I(\alpha)$ returns one if $\alpha$ is true and zero otherwise, for an arbitrary boolean input $\alpha$.

$j$; an index over features.

$J$; the total number of features under consideration.

$k$; an index over ICUs.

$k$; a set of simulation parameters.

$k^4$; the class of simulation parameter sets for which $k_3 = 2.5$ and $k_2^2/k_1^2 = 4$.

$k^{19}$; the class of simulation parameter sets for which $k_3 = 2.5$ and $k_2^2/k_1^2 = 19$.

$k^{B4}$; the set of simulation parameter values from within $k^4$ that best match $W^B$.

$k^{B19}$; the set of simulation parameter values from within $k^{19}$ that best match $W^B$.

$k^{P4}$; the set of simulation parameter values from within $k^4$ that best match $W^P$.

$k^{P19}$; the set of simulation parameter values from within $k^{19}$ that best match $W^P$.

$K$; the total number of ICUs under consideration.

$l_i$; the Specific Component associated with the $i$'th patient. This is a measure of the extent to which the $i$'th patient is more unwell than an average patient from the same ICU.

$\hat{l}_i$; an estimate for the Specific Component associated with the $i$'th patient.

$L_{m}$; the Specific Ability of the $m$'th MPM. This is a measure of the extent to which the given MPM can account for the extent to which a given patient is more unwell than an average patient treated in the same ICU.
\( \hat{L} \); the likelihood function.

\( m \); an index over the \( MPMI^1 \)'s within a given \( MPMI^2 \).

\( M \); the total number of \( MPMI^1 \)'s contained within a given \( MPMI^2 \).

\( \text{max} \); a function that outputs the maximum value of the input vector.

\( MVN \); a multivariate normal function. \( MVN(\alpha; \mu, \sigma) \) is the probability density function for a multivariate normal distribution with vector of means \( \mu \) and covariance matrix \( \sigma \) evaluated at \( \alpha \).

\( n \); the Null Component. The logit of the average mortality rate for an average patient treated in an average ICU.

\( N \); a normal distribution. \( N(\mu, \sigma) \) represents a univariate normal distribution with mean \( \mu \) and variance \( \sigma \).

\( p_i \); the \( i \)'th patient’s risk. The probability that the \( i \)'th patient will die before hospital discharge.

\( \hat{p}_i \); a risk estimate for the \( i \)'th patient under consideration (usually generated by an MPM).

\( P \); a function that outputs the probability of its argument being true.

\( q_k \); the quality of care provided by the \( k \)'th ICU under consideration.

\( r_i \); the logit of the \( i \)'th patient’s risk.

\( \hat{r}_i \); the logit of a risk estimate for the \( i \)'th patient under consideration (usually generated by an MPM).

\( \text{SLL} \); the standardized log-likelihood, an MPM performance evaluator (defined in Subsection 5.4.1).

\( t_i \); the prediction time for the \( i \)'th patient or ICU stay under consideration.

\( T_i \); the total duration of the \( i \)'th ICU stay under consideration.

\( w \); an index over MPM evaluators.

\( W \); a set of risk array statistics. The application of \( W \) yields 9 values that summarize the properties of the input risk array.
$W^B$; the output from the evaluation of the risk array statistics on the Broad $MPMI^1$ Subset of the Representative MPMC.

$W^P$; the output from the evaluation of the risk array statistics on the Performance $MPMI^1$ Subset of the Representative MPMC.

$x$; a design matrix. $x_{ik}$ provides the value of the $k$’th feature for the $i$’th patient under consideration.

$y_i$; the measured response (usually death before hospital discharge) for the $i$’th patient under consideration.

$\hat{y}_i$; an estimate for the response of the $i$’th patient under consideration.

$z$; a partitioning matrix.

$Z^k$; the mean performance of the MPMs selected by a given MPM evaluator, evaluated using simulation parameter set $k$. 
Appendix C

The Standard Preprocessor Inducer

The ‘Standard Preprocessor’ was introduced in Section 3.6 as a class of methods for replacing missing feature values and ensuring that feature values are somewhat normally distributed, while the ‘Standard Preprocessor Inducer’ was introduced as a method for generating a Standard Preprocessor. The steps applied by the Standard Preprocessor Inducer and the form of the Standard Preprocessors that it outputs are detailed in the present Appendix. Examples of how the Standard Preprocessor transforms features are provided in Subsection 3.7.1 and Appendix D.

Numeric Features

Let \( \Gamma \) be a function that takes as inputs a single numeric feature, \( x \), and a set of four parameters, \( \{ \beta_1, \beta_2, \beta_3, \beta_4 \} \), and that outputs a feature through the application of the following steps:

1. For \( i \in \{1,2,\ldots,I\} \):
   
   (a) If \( x_i > \beta_1 \) set \( x_i \) equal to \( \beta_3 \).
   
   (b) If \( x_i < \beta_2 \) set \( x_i \) equal to \( \beta_3 \).
   
   (c) If \( x_i \) is missing set \( x_i \) equal to \( \beta_3 \).
2. If $\beta_i > 0$

   (a) Set $\theta$ equal to the median of the non-zero elements of $x$.

   (b) Set $x$ equal to $\log(x + (\beta_i - 1)^2 \theta / 100)$.

3. Return $x$.

Step 1 of $\Gamma$ removes outliers and sets all missing values equal to $\beta_3$. Step 2 of $\Gamma$ is optional and allows the feature space to undergo one of a family of logarithmic transformations.

**Categorical Features**

Let $\Theta$ be a function that takes as inputs a single categorical feature, $x$, and a list of categories, $\{\alpha_1, \alpha_2, \ldots, \alpha_\Psi\}$, and that outputs a set of features through the application of the following steps:

1. Create an $I \times \Psi$ matrix of zeros, $\tau$.

2. For every combination of $\psi \in \{1, 2, \ldots, \Psi\}$ and $i \in \{1, 2, \ldots, I\}$: If $x_i = \alpha_\psi$ set $\tau_{i,\psi}$ equal to unity.

3. Return $\psi$.

The action of $\Theta$ is to split a single categorical feature into a set of one or more binary features, with the value of the $\psi$'th such binary feature for the $i$'th patient indicating whether or not the value of the input feature for that patient was equal to the $\psi$'th category.

**The Standard Preprocessor**

Let $\Omega$ denote a version of the Standard Preprocessor that can be applied to a design matrix with $K$ features of a particular types. Then $\Omega$ takes as inputs an $I \times K$ design matrix, $x$, and a list of parameter sets of length $K$, $\{\Phi_1, \Phi_2, \ldots, \Phi_K\}$, and proceeds through the following steps:
1. Create an $I \times 0$ matrix, $\rho$.

2. For $k \in \{1, 2, \ldots, K\}$:

   (a) If the $k$’th column of $x$ is numeric set $\sigma$ to be the output from the application of $\Gamma$ with parameters $\Phi_k$ to the $k$’th column of $x$.

   (b) If the $k$’th column of $x$ is categorical set $\sigma$ to be the output from the application of $\Theta$ with parameters $\Phi_k$ to the $k$’th column of $x$.

   (c) Add $\sigma$ to the end of $\rho$ column-wise.

3. Return $\rho$.

The Standard Preprocessor therefore corresponds to an application of $\Gamma$ to each numeric feature in $x$ and an application of $\Theta$ to each categorical feature in $x$, with the output of the Standard Preprocessor corresponding to the union of all features output by each of these individual applications of $\Gamma$ and $\Theta$.

**Normality Evaluation**

Let $\Upsilon$ be a function that takes as inputs a single numeric feature, $x$, and that outputs a measure of the extent to which the input feature requires further preprocessing, through the application of the following steps:

1. Set $\gamma$ equal to the number of entries in $x$ that were originally non-missing.

2. Set $\phi$ equal to the number of imputed entries in $x$.

3. Set $\xi$ to be a vector containing the non-imputed entries from $x$.

4. Set $\xi = \frac{\xi - E(\xi)}{\sqrt{E(\xi)}}$.

5. Set $\iota$ equal to $\phi \times log(0.01/\gamma) - \sum \sigma^2$. 

6. Return $\tau$.

For a given numeric feature, $\Upsilon$ applies two types of penalties: one for having feature values that are far from normally distributed and one for having too many imputed values. The balance between the two penalties was chosen such that the set of outlier removals required to minimize the output of $\Upsilon$, when it is applied to a feature the values of which are drawn from a normal distribution, is the empty set exactly 99% of the time.

**The Standard Preprocessor Inducer**

The Standard Preprocessor Inducer takes as inputs an $I \times K$ design matrix, $x$, and outputs a Standard Preprocessor according to the following steps:

1. Let $\Xi$ be an empty list of parameter sets.

2. For $k \in \{1, 2, \ldots, K\}$:
   
   (a) If the $k$’th column of $x$ is numeric
      
      i. Let $\xi$ denote the set of length four vectors containing all possible combinations of $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ for which $\beta_1$, $\beta_2$ and $\beta_3$ are real numbers and $\beta_4 \in \{0, 1, 2, \ldots, 10\}$.
      
      ii. Let $\mu$ denote the subset of vectors in $\xi$ for which $\Upsilon(\Gamma(x, \{\beta_1, \beta_2, \beta_3, \beta_4\}))$ takes its minimum value.
      
      iii. Have $\mu$ retain only vectors for which $\beta_1$ takes its minimal value.
      
      iv. Have $\mu$ retain only vectors for which $\beta_2$ takes its maximal value.
      
      v. Set $\Xi_k$ equal to the vector in $\mu$ for which $\beta_4$ takes its minimal value.
   
   (b) If the $k$’th column of $x$ is categorical set $\Xi_k$ equal to the set of categories in $x$ that contain at least $I/200$ entries.

3. Return $\Xi$. 
The Standard Preprocessor Inducer therefore outputs the Standard Preprocessor that modifies $x$ such that the outputs from the application of $\Upsilon$ to each feature are minimized.
Appendix D

Further Normal Probability Plots

The Standard Extractor was introduced in Subsection 3.4.2 as a method for generating a small and well-defined set of, primarily, demographic and physiological features that are based on the features required to evaluate the APACHE IV model. In Section 3.6 the ‘Standard Preprocessor’ was introduced (fully described in Appendix C) that takes a feature and applies to it a combination of transformations and value-discarding so as to make the feature more normally distributed, ready for the learner inducers of Chapter 4. Subsection 3.7.1 included normal probability plots for each of temperature, white blood count and Glasgow Coma Scale (GCS), both before and after the application of the Standard Preprocessor. This Appendix extends the normal probability plots to include the remaining 17 numeric features from the Standard Extractor that were not presented in Subsection 3.7.1.

Feature distributions before and after the application of the Standard Preprocessor are presented as normal density plots in Figures D.1, D.2, D.3, D.4, D.5 and D.6. Only the features that have not already been dealt with in Figure 3.2 are considered in this Appendix. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. If a feature is approximately normally distributed then its points on a normal probability plot should approximate a straight line. The features considered in this Appendix are in no particular order.
Fig. D.1 Normal probability plots for mean arterial blood pressure (left panels), heart rate (center panels) and respiratory rate (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Fig. D.2 Normal probability plots for urine output (left panels), white hematocrit (center panels) and blood sodium level (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Fig. D.3 Normal probability plots for blood urea nitrogen levels (left panels), creatinine (center panels) and blood glucose level (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Fig. D.4 Normal probability plots for albumin (left panels), white bilirubin (center panels) and $PaO_2/FiO2$ (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Fig. D.5 Normal probability plots for partial pressure of oxygen in the blood (left panels), partial pressure of carbon dioxide in the blood (center panels) and blood pH (right panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Fig. D.6 Normal probability plots for partial pressure of age (left panels) and time between hospital admission and ICU admission (center panels) before (top panels) and after (bottom panels) the application of the Standard Preprocessor. A normal probability plot plots the quantiles of a feature against the equivalent quantiles of a normal distribution. Points in red represent feature values that were discarded by the Standard Preprocessor.
Appendix E

Risking vs Benchmarking Feature Importance

Given a set of features and an MPM evaluator, there are various methods with which one can assess feature importance. One simple method is to first train and evaluate the performance of an MPM using all of the features. For each feature a new MPM is then trained and evaluated using all but the feature under consideration. The feature importance can then be evaluated as the performance of the complete model minus the performance of the model trained without the feature under consideration. An alternative method is to train an MPM using each single feature in turn and then compare the performance of these MPMs with the Null Model, with the difference again providing a measure of feature importance. In this Appendix the latter approach is used to measure feature importance according to each of the SLL and $\chi^2$ in turn, such that the differences between these two measures of MPM performance can be investigated further. The latter approach is favored because it does not mask the importance of features that are highly correlated with one another.

An $MPMI^2$ is defined that contains 27 $MPMI^1$s. The $m$'th $MPMI^1$ for $m \in \{1, 2, \ldots, 26\}$ is set to be RLR using only the $m$'th Standard Feature, as listed in Table 3.4, while the 27th
MPMI\textsuperscript{1} is set to be the Null model. An MPMC is then applied to the APACHE Database with this MPMI\textsuperscript{2} and options (2.1,5.2,6.1,10.6,13.2,14.1), as defined in Table 5.4.

The performance point estimates and standard error estimates output by the MPMC for each of the SLL and $\chi^2$ are presented in Figure E.1. Features at the bottom of the plot\textsuperscript{1} exhibit good risking performance, while features at the right exhibit good benchmarking performance.

Unsurprisingly, the inclusion of almost any feature results in an MPM that provides better risking performance than the Null Model. The only exception is the feature ‘Unable’, which is a binary flag that states whether or not the GCS could be measured for the given patient. The importance of this feature is potentially limited here because it is not being considered in conjunction with GCS. The most important features for risking, according to this measure at least, are the GCS and Diagnosis Group, precisely as one might expect.

The plot clearly exhibits a positive gradient, indicating that features that are important for risking also tend to be important for benchmarking. However, the correlation exhibited in Figure E.1 is quite limited. The extent to which the points deviate from a straight line far exceeds the bounds associated with their standard error estimates. For example, the location of GCS to the left and far below Prior Location implies that, although the latter is far less important in terms of risking, it is more important in terms of benchmarking.

\textsuperscript{1}Orientated such that the title of the plot is at the far left
Fig. E.1 Importance plot for the Standard Features evaluated on the APACHE Dataset. The x and y coordinates of the center of a given cross provides the mean risking and benchmarking performance, respectively, for MPMs trained using only the given feature. The size of a given cross represents the associated standard error estimates. The null model is marked in blue to make it easily distinguishable.