Dirichlet process probit
misclassification mixture model for
misclassified binary data

Zhenzheng (Helen) Hu

SN: 17102367

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Department of Statistical Science
University College London

December 3, 2021
Declaration of authorship

I, Zhenzheng Hu, hereby declare that I have written this thesis without any help from others and without the use of documents and aids other than those stated above.
Acknowledgements

I would like to thank the Alan Turing Institute for funding my PhD studies and my supervisors Dr. Ioanna Manolopoulou and Dr. Ioannis Kosmidis for their supervision. They have been incredibly kind, supportive throughout my PhD and I have learned a great deal from them. I would finally like to thank friends and family for their support.
Abstract

Mislabelling or misclassification in binary data refers to incorrectly labelled responses and could arise due to problems in the labelling process or imperfect evidence for labelling. The latent misclassification process could take a variety of forms depending on how it relates to the true labels as well as the associated covariates of each response. Modelling under misclassification is challenging because of the inherent identifiability issues and ignoring misclassification could lead to inaccurate inferences. Statistical methods addressing misclassification have appeared in the literature in a variety of contexts, sometimes using different terminology, and often focusing on a particular application. In this thesis, we first cast existing statistical methods under a unified framework and later propose a new flexible Bayesian mixture model for modelling misclassified binary data - the Dirichlet process probit misclassification mixture model. The main idea is to assume a Dirichlet process mixture model over the covariate space and misclassification probabilities. This naturally partitions observations into clusters where different clusters can possess different misclassification probabilities. The clustering uses both covariates and observed responses and covariates are approximated using a Dirichlet mixture of multivariate Gaussians. The incorporation of cluster-specific misclassification probabilities takes into consideration of the misclassification in the observed responses. An efficient Gibbs-like algorithm is available based on the truncated approximation of Dirichlet process and the stick-breaking construction.

This thesis is motivated by the pervasiveness of label noise in a wide variety of applications, coupled with the lack of unified statistical exposition and comparison of all available methods. The structure of the thesis as follows. Chapter 1 introduces the problem of label misclassification and reviews existing methods for modelling misclassification in binary data.
Chapter 2 discusses the basic of Bayesian nonparametrics, Dirichlet process, Dirichlet process mixture models, and posterior inference procedures for Dirichlet process mixture models, which are essential components of the Dirichlet process probit misclassification mixtures that we propose later. Chapter 3 describes our proposed model for modelling mislabelled binary data. Chapter 4 presents experimental studies on our proposed model using a real dataset. Section 5 wraps up the discussion on the topic and include final remarks such as possible model extension.
Impact statement

This dissertation addresses a common challenge in binary studies: observations which are potentially mislabelled but whose mislabelling does not occur randomly. The issues of mislabelling arise very frequently, especially in big data contexts such as crowd-sourced data, data collected without the opportunity of experimental design (e.g. over the internet), or which rely on self-reporting (e.g. via wearable devices, apps in mobile platforms or questionnaires). Ignoring misclassification could lead to inaccurate inferences and predictions.

This first chunk of this thesis provides the first systematic review of current methods for dealing with misclassification. This could benefit researchers and practitioners facing the problem of misclassification at hand. This thesis also proposes a flexible method that models misclassification without assuming constant misclassification probability nor the availability of validation data. The proposed method can be applied to any binary data prone to contamination in the responses regardless of application area. For example, in a self-reported survey where user behaviours such as smoking are asked, the responses are likely to be inaccurate to a different degree from person to person. In another context where we have a medical dataset where the disease outcomes are measured by different tests, we expect different amounts of false positives and false negatives among the observations measured different tests. In both scenarios, misclassification likely do not occur at random across observations and our proposed method can be directly applied to the observed contaminated data to inference the true responses that are of interest.
Contents

1 Modelling misclassification in binary data ........................................ 17

1.1 Introduction .................................................................................. 18

1.2 General framework ....................................................................... 21

1.2.1 Notation .................................................................................. 21

1.2.2 Likelihood approach ................................................................. 23

1.3 Known misclassification probabilities ........................................... 24

1.3.1 Expectation-Maximisation ....................................................... 24

1.3.2 Post-fit adjustment of regression coefficients ........................... 27

1.3.3 Misclassification Simulation Extrapolation ............................... 29

1.3.4 Maximum estimated likelihood ................................................ 33

1.4 Modelling unknown misclassification probabilities by assuming validation data 35

1.4.1 Two-step modelling ................................................................. 36

1.4.2 Full Bayesian parametric modelling ......................................... 38

1.4.3 Multiple imputation and maximum likelihood .......................... 40

1.5 Other methods ............................................................................. 43

1.5.1 Bayesian modelling using a prior on misclassification probabilities 43

1.5.2 Non model-based approaches ................................................. 44

1.5.3 Summary .................................................................................. 45

1.6 Empirical evaluation and comparison of methods .......................... 47
1.6.1 Set up ......................................................... 48
1.6.2 Implementation details ..................................... 51
1.6.3 Evaluation criteria ......................................... 54
1.6.4 Results ....................................................... 55
1.6.5 Advantages and disadvantages .............................. 72

2 Bayesian Nonparametrics ........................................ 74
  2.1 Notations and useful definitions ............................. 75
  2.2 Gamma distribution ......................................... 77
  2.3 Beta distribution ............................................. 77
  2.4 Dirichlet distribution ........................................ 78
    2.4.1 Conjugacy with the multinomial distribution ............ 79
    2.4.2 Other properties ........................................ 80
  2.5 Dirichlet process ........................................... 81
    2.5.1 Conjugacy property ..................................... 82
    2.5.2 Sethuraman’s stick-breaking construction .............. 84
  2.6 Applications of Dirichlet process .......................... 86
    2.6.1 Mixture model with symmetric Dirichlet prior .......... 86
    2.6.2 Dirichlet process mixture model ....................... 89
    2.6.3 Density estimation ...................................... 91
  2.7 Posterior inference for Dirichlet process mixture model .... 92
  2.8 Basics of Monte Carlo simulation .......................... 92
    2.8.1 Gibbs sampler .......................................... 93
    2.8.2 Metropolis and Metropolis-Hastings algorithm .......... 94
  2.9 Markov chain Monte Carlo methods for Dirichlet process mixture model .... 95
    2.9.1 Truncated Gibbs sampler ................................ 96
3 Dirichlet process probit misclassification mixture model for misclassified binary data  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Model structure</td>
<td>102</td>
</tr>
<tr>
<td>3.2 Model priors</td>
<td>105</td>
</tr>
<tr>
<td>3.3 Simple example</td>
<td>107</td>
</tr>
<tr>
<td>3.3.1 Simulated data</td>
<td>107</td>
</tr>
<tr>
<td>3.3.2 DP-PMM outputs</td>
<td>109</td>
</tr>
<tr>
<td>3.3.3 Prediction using DP-PMM</td>
<td>112</td>
</tr>
<tr>
<td>3.4 Posterior inference for proposed model</td>
<td>113</td>
</tr>
<tr>
<td>3.4.1 Posterior of cluster-specific parameters $\theta$</td>
<td>113</td>
</tr>
<tr>
<td>3.4.2 Posterior of clusters allocation parameters $\pi$ and $z$</td>
<td>115</td>
</tr>
<tr>
<td>3.4.3 Posterior of model parameters for responses $\beta$ and $t$</td>
<td>117</td>
</tr>
<tr>
<td>3.5 Predictive inference</td>
<td>122</td>
</tr>
</tbody>
</table>

4 Experiment on real data  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Set up</td>
<td>124</td>
</tr>
<tr>
<td>4.1.1 Injection of misclassification</td>
<td>127</td>
</tr>
<tr>
<td>4.1.2 Priors used</td>
<td>128</td>
</tr>
<tr>
<td>4.1.3 Gibbs sampling</td>
<td>129</td>
</tr>
<tr>
<td>4.1.4 Consideration on label switching</td>
<td>129</td>
</tr>
<tr>
<td>4.2 Evaluation criteria</td>
<td>130</td>
</tr>
<tr>
<td>4.2.1 Criteria for inferential ability</td>
<td>130</td>
</tr>
<tr>
<td>4.2.2 Criteria for predictive ability</td>
<td>131</td>
</tr>
<tr>
<td>4.3 Results and discussions</td>
<td>132</td>
</tr>
<tr>
<td>4.3.1 Case 1: no misclassification</td>
<td>132</td>
</tr>
</tbody>
</table>
4.3.2 Case 2: low misclassification ................................. 135
4.3.3 Case 3: moderate misclassification .............................. 141
4.3.4 Case 4: high misclassification ................................. 146
4.3.5 Further comparison of methods ................................. 150
4.3.6 Further consideration on DP-PMM ............................... 152

5 Final remarks .................................................. 154
5.1 Future directions ................................................. 155
  5.1.1 Extend the link function beyond probit link .................. 155
  5.1.2 Automatic misclassification type selection .................. 156
  5.1.3 Incorporation of validation data .............................. 157
  5.1.4 Applications ................................................. 157

A Supplementary material for methods from literature review .......... 159
  A.1 Derivations of the magnitude of bias ............................ 159
  A.2 Adding misclassification in MC-SIMEX ............................ 161
  A.3 Properties of MEL estimator and computing robust distance .. 163
  A.4 Multiple imputation and Rubin’s rule ............................ 167

B Existence of maximum likelihood estimator in logistic regression .... 171
  B.1 Conditions for the existence of maximum likelihood estimator in logistic regression models ........................................... 171
  B.2 Identification issue ............................................. 176
  B.3 Firth’s correction for maximum likelihood estimator ............ 178

C Distribution review .............................................. 179
  C.1 Additional notes on Beta distribution ............................ 179
  C.2 Wishart distribution ............................................. 180
C.3 Normal Inverse Wishart (NIW) ........................................... 180
C.4 Conjugacy with MVN .................................................... 180
  C.4.1 Conjugacy derivation for the Normal part ....................... 182
  C.4.2 Conjugacy derivation for the Inverse Wishart part ............ 184
  C.4.3 Sampling of normal inverse wishart distributed variable .... 184
  C.4.4 Updating of normal and inverse wishart parameter in each cluster . 185

D Additional discussion on Dirichlet process and Dirichlet process mixture models 187
D.1 Black-MacQueen urn scheme ........................................ 187
D.2 Chinese Restaurant Process ......................................... 188
D.3 Marginalised Gibbs sampler ....................................... 190

E Additional materials on proposed DP-PMM 192
E.1 Auxiliary variable for Bayesian probit model .................... 192
E.2 Label Switching ....................................................... 194
  E.2.1 Equivalence class representation algorithm .................... 195
  E.2.2 Data-based algorithm ........................................... 197
  E.2.3 Label switching algorithm consideration for DP-PMM ........ 197
E.3 $t$-distribution approximation for Bayesian logistic regression .... 198
List of Figures

1.1 Different types of misclassification according to different taxonomies in Statistics and machine learning literature where the probability of misclassifying the $i$th observation is either $\gamma_{0i} = p(y_i = 1 | t_i = 0, x_i)$ or $\gamma_{1i} = p(y_i = 0 | t_i = 1, x_i)$; $t_i$ and $y_i$ are the true response and observed response of the $i$th observation, $x_i$ are the observed covariates and $v_i$ are the covariates that determine the misclassification probabilities, which can include $x_i$, but may also contain, for example, additional variability. ........................................ 20

1.2 Data generation process that illustrates the relationship between covariates $X$, true responses $t$, and observed responses $y$. In simulated datasets, we define a distribution on covariates $X$ and generate true responses $t$ conditionally on $X$ through a logistic regression model. In phoneme data, we take responses at hand as true responses. Individual positive and negative labels of $t$ are then ‘flipped’ with probability $\gamma_1$ and $\gamma_0$ respectively to generate observed responses $y$. ........................................................................... 48

1.3 Comparison of the estimates ($\hat{\beta}$ denotes average) of $\beta_0$ in simulated dataset in exp 1 where the proportion of 1/0 are roughly balanced with $(\beta_0, \beta_1, \beta_2)_{true} = (-0.4, 0.4, 0.4)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters. 57
1.4 Comparison of the estimates (◊ denotes average) of $\beta_1$ in simulated dataset in exp 1 where the proportion of 1/0 are roughly balanced with $(\beta_0, \beta_1, \beta_2)^{true} = (-0.4, 0.4, 0.4)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.

1.5 Comparison of the estimates (◊ denotes average) of $\beta_1$ in simulated dataset in exp 2 where the proportion of 1/0 are roughly 20/80 with $(\beta_0^{true}, \beta_1^{true}, \beta_2^{true}) = (-1.4, -2.2, -1)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.

1.6 Comparison of the estimates (◊ denotes average) of $\beta_2$ in simulated dataset in exp 2 where the proportion of 1/0 are roughly 20/80 with $(\beta_0^{true}, \beta_1^{true}, \beta_2^{true}) = (-1.4, -2.2, -1)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.

1.7 Comparison of the estimates (◊ denotes average) of $\beta_0$ in phoneme dataset experiment where the proportion of 1/0 are roughly 30/70 with $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (-1.065, -0.610, -0.406, 0.672, 0.788, 0.541)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.

1.8 Comparison of the estimates (◊ denotes average) of $\beta_4$ in phoneme dataset where the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.

1.9 Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in exp 1 using various methods where the data are simulated and the proportion of 1/0 are roughly 50/50 and misclassified responses generated from true responses under different $(\gamma_0, \gamma_1)$. The boxplots represent the accuracy and F-measure from 1000 runs and bars represent their standard deviations.
1.10 Comparison of AUC evaluated on test data in exp 1 using various methods where the data are simulated and the proportion of 1/0 are 50/50 and misclassified responses are generated from true responses under different $\gamma_0, \gamma_1$. The boxplots represent the AUC from 1000 runs and bars represent their standard deviations.

1.11 Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in exp 2 using various methods where the data are simulated and the proportion of 1/0 are 20/80 and misclassified responses are generated from true responses under different $\gamma_0, \gamma_1$. The boxplots represent the accuracy and F-measure from 1000 runs and bars represent their standard deviations.

1.12 Comparison of AUC evaluated on test data in exp 2 using various methods where the data are simulated and the proportion of 1/0 are 20/80 and misclassified responses are generated from true responses under different $\gamma_0, \gamma_1$. The boxplots represent the AUC from 1000 runs and bars represent their standard deviations.

1.13 Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in phoneme dataset experiment using various methods where the data are simulated and the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true responses under different $\gamma_0, \gamma_1$. The boxplots represent the accuracy and F-measure from 100 runs and bars represent their standard deviations.

1.14 Comparison of AUC evaluated on test data using various methods in phoneme dataset experiment where the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true ones under different $\gamma_0, \gamma_1$. The boxplots represent the AUC from 100 runs and bars represent their standard deviations.
3.1 Graphical model representation of data generating process (shaded circles denote observed variables and unshaded circles denote unobserved variables and parameters); $\theta^* = (\mu^*, \Sigma^*, \gamma^*)$ represents cluster-specific Gaussian mean, variance, and misclassification probabilities. $n$ denotes the size of the data and $K$ represents the maximum number of clusters in the data.

3.2 Visualisation of the true responses (left) and observed contaminated responses (right) in training data where the green line indicates the decision boundary for the true responses.

3.3 Visualisation true membership of observations to the Gaussian components (left) and the best cluster assignment (right) in simulated dataset.

3.4 Heat map of posterior similarity matrix of clustering on training data of simulated dataset.

3.5 Histogram of average $\gamma^*_0$ in nonempty clusters (left) and histogram of average $\gamma^*_1$ in nonempty clusters (right).

3.6 Visualisation of model output: estimated regression coefficients for probit model with vertical lines indicating the 95% credible interval (left) and the number of nonempty clusters in all iterations (right).

3.7 True responses of test data (left) average probability of 1 for test data using estimated probit regression coefficients from all 5000 MCMC iterations (right).

4.1 Histogram of each covariates and responses at hand $t$.

4.2 Pairwise plot of covariates for phoneme dataset with colour-coded true responses (pink for 0 and green for 1).

4.3 Traceplots for concentration parameter $\alpha$ (left) and $\beta_0$ (right).
4.4 95% credible interval (black line) for estimated $\beta_j$ from DP-PMM in the phoneme dataset when there is no misclassification injected compared to ‘true values’ from probit regression $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ indicated by red line. .................................................. 133

4.5 Histogram of average $\gamma_{0h}^*$ in all clusters (left) and histogram of average $\gamma_{1h}^*$ in all clusters (right) from DP-PMM in the phoneme dataset when there is no misclassification injected in training data. ............................................. 133

4.6 Histogram of number of nonempty clusters among 5000 MCMC runs when there is no misclassification injected in the training data of phoneme dataset. 134

4.7 Heat map of posterior similarity matrix of clustering on phoneme training data when there is no misclassification injected. .................................................. 134

4.8 Misclassification probabilities a a function of $X_5$ in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset. .................................................. 135

4.9 Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{0i}$ (right) for all observation $i$ in phoneme training data when there is low misclassification. Blue line indicates the Beta prior used for $\gamma_{0h}^*, \gamma_{1h}^*$ in DP-PMM. .................................................. 136

4.10 95% credible interval for estimated $\beta_j$ from DP-PMM fitted phoneme training data with low misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.31$ from probit regression fitted to the uncontaminated training data indicated by the red lines. .................................................. 137

4.11 Histogram of estimated average misclassification probabilities for all clusters from DPMM in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset. .................................................. 138
4.12 Estimated mean misclassification probabilities as a function $X_3$ in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.

4.13 Heatmap of posterior similarity matrix for clustering results from DPMM in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.

4.14 Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{1i}$ (right) for all observation $i$ in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset. Blue line indicates the Beta prior used for $\gamma_{0h}^*, \gamma_{1h}^*$ in DP-PMM.

4.15 95% credible interval for estimated $\beta_j$ from DP-PMM fitted to phoneme training data with moderate level of misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ from probit regression fitted to the uncontaminated training data indicated by the red lines.

4.16 Histogram of estimated average misclassification probabilities for all clusters from DPMM in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset.

4.17 Estimated mean misclassification probabilities as a function $X_4$ in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset.

4.18 Heatmap of posterior similarity matrix for clustering results from DPMM in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset.
4.19 Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{0i}$ (right) for all observation $i$ in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset. Blue line indicates the Beta prior used for $\gamma_{0h}, \gamma_{1h}$ in DP-PMM. .................................................. 146

4.20 95% credible interval for estimated $\beta_j$ from DP-PMM fitted to phoneme training data with high level of misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ from probit regression fitted to the uncontaminated training data indicated by the red lines. .............................. 147

4.21 Histogram of estimated average misclassification probabilities for all clusters from DPMM in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset. ................................. 148

4.22 Estimated mean misclassification probabilities as a function $X_4$ in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset. ................................. 148

4.23 Heatmap of posterior similarity matrix for clustering results from DPMM in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset. ................................. 149

4.24 Histogram of number of nonempty clusters among 5000 MCMC runs when there is no misclassification injected in the training data of phoneme dataset and misclassification probabilities are set to be 0 for all observations. ................................. 153

B.1 Sample data points patterns - top: completely separable, overlap; bottom: two cases of quasi-separable. .................................................. 176
List of Tables

1.1 Summary of methods described in detail in this thesis where MEL is maximum estimated likelihood, and ML is maximum likelihood. The first four methods assume known non-differential misclassification probability while the last three methods require validation data whose true responses are known in addition to the observed responses. ........................................... 21

1.2 Summary of available implementation tools for methods described in detail in this paper. Summary of methods described in detail in this paper where MEL is maximum estimated likelihood, ML is maximum likelihood. .............. 52

1.3 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2$ under different $(\gamma_0, \gamma_1)$ when the proportion of 1/0 are roughly balanced with $(\beta_0, \beta_1, \beta_2)^{true} = (-0.4, 0.4, 0.4)^T$ and experiments are repeated 1000 times under pre-determined seeds. ...... 59
1.4 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2$ in exp 2 under different $(\gamma_0, \gamma_1)$ when the proportion of 1/0 are roughly 20/80 with $(\beta_0, \beta_1, \beta_2)^{true} = (-1.4, -2.2, -1)^T$ and experiments are repeated 1000 times under pre-determined seeds. Note that even though the median estimated values for $\beta$’s using the two-step modelling seem to have extreme mean estimates, the median estimates are actually close to true values as shown in Figure 1.6. The large mean estimates are due to separation issue in validation data and we use the \texttt{brglm2} package of Kosmidis (2017) to learn the logistic regression model for $p(y_i = 1 \mid t_i, x_i)$ in validation data. The use of the bias-reducing adjusted scores method avoids infinite estimates but sometimes give large values of estimated $\beta$’s.

1.5 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ under different $(\gamma_0, \gamma_1)$ for the phoneme dataset where the proportion of 1/0 are roughly 70/30, $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (-1.065, -0.610, -0.406, 0.672, 0.788, 0.541)^T$ and experiments are repeated 1000 times under pre-determined seeds.

4.1 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ in phoneme dataset under low misclassification setting where $(\bar{\gamma}_0, \bar{\gamma}_1) = (0.022, 0.030)$. The result from DP-PMM refers to the mean of estimated $\beta$ over 5000 MCMC runs.

4.2 Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under low misclassification setting where $(\bar{\gamma}_0, \bar{\gamma}_1) = (0.022, 0.030)$. Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated $\beta$ in every 200 run out of 5000 MCMC runs in total.
4.3 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ in phoneme dataset under moderate misclassification setting where $(\tilde{\gamma}_0, \tilde{\gamma}_1) = (0.066, 0.091)$. The result from DP-PMM refers to the mean of estimated $\beta$ over 5000 MCMC runs.

4.4 Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under moderate misclassification setting where $(\tilde{\gamma}_0, \tilde{\gamma}_1) = (0.066, 0.091)$. Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated $\beta$ in every 200 run out of 5000 MCMC runs in total.

4.5 Comparison of the average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ in phoneme dataset under high misclassification setting where $(\tilde{\gamma}_0, \tilde{\gamma}_1) = (0.132, 0.169)$. The result from DP-PMM refers to the mean of estimated $\beta$ over 5000 MCMC runs.

4.6 Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under high misclassification setting where $(\tilde{\gamma}_0, \tilde{\gamma}_1) = (0.132, 0.169)$. Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated $\beta$ in every 200 run out of 5000 MCMC runs in total.
Chapter 1

Modelling misclassification in binary data

This chapter introduces the problem of label misclassification in binary regression. In Section 1.1, we discuss the causes and impact of misclassification and then present a taxonomy of the methods that try to alleviate the consequences of misclassification. In Section 1.2 we introduce the general framework. We then give a comprehensive overview of various model-based methods for inference under known non-differential misclassification probabilities in Section 1.3. In Section 1.4 we review methods which do not assume known misclassification probabilities, but instead impose other assumptions, such as the availability of a validation dataset containing both true and observed responses. A list of all methods described in detail in this thesis can be found in Table 1.1 and Section 1.5.2 gives a brief summary of non model-based methods where misclassification are not modelled explicitly. We wrap up the discussion of this chapter by providing an extensive experimental comparison of existing methods described earlier on simulated data and real dataset in Section 1.6 and discuss the advantages and disadvantages of each method.
1.1 Introduction

Mislabelled responses or misclassification has been a common challenge in many studies that involve binary responses. In the presence of mislabelling, the observed binary responses may be false 1 and false 0 with positive probabilities. For example, in medical studies, mislabelled outcomes can occur because of inaccurate tests (Pechenizkiy et al., 2006). Errors could also be attributed to clerical error in recording the data, or to the subjective nature of labelling prone to inter-expert variability. An example of the latter type is disagreement in medical diagnostic tests such as the boundaries of signal patterns in electrocardiogram analysis (Hughes et al., 2004). Misclassification also frequently occurs in studies where assorted outcomes are collected through self-reported surveys, where the outcomes cannot be trusted with complete accuracy (Pérez et al., 2007). The recent trend of crowd-sourcing data further normalises misclassification since such data are often unverified and prone to error (Howe, 2006; Sheng et al., 2008).

In the statistics literature, early reports of the issue of mislabelling appeared in studies in medical statistics and epidemiology more than half century ago, such as Bross (1954) and Harper (1964). Ignoring misclassification in responses may lead to inaccurate inference and model predictions and estimates of poor quality (Neuhaus, 1999; Zhu and Wu, 2004; Cannings et al., 2020). Specifically, the consequences of ignoring misclassification include: a decrease in classification performance (Natarajan et al., 2013); bias and efficiency loss in the estimation of parametric models (Neuhaus, 1999, 2002); changes in learning requirements such as an increase in the number of observations needed to achieve the desired level of accuracy (Angluin and Laird, 1988; Gentile and Helmbold, 2001); increase in complexity of learned models, e.g. more nodes in decision trees (Frénay and Verleysen 2013; Oza 2004); distortion of observed frequencies (and, hence, incorrect estimates of prevalence) (Bross, 1954; Frénay and Verleysen, 2013); added difficulties for feature selection (Gerlach and Stamey,
Label misclassification can be categorised in a variety of ways as illustrated in Figure 1.1. In the review of Frénay and Verleysen (2013), three distinct types of misclassification are presented: misclassification completely at random, misclassification at random, and misclassification not at random, mirroring the types of missingness in the missing data literature (Rubin, 1976; Schafer and Graham, 2002). *Misclassification completely at random* describes the situation where a 1 or 0 has the same fixed probability of being misclassified across all observations. *Misclassification at random* refers to the case where the probability of misclassifying a 1 and the probability of misclassifying a 0 are constant but need not be equal. In other words, *misclassification at random* implies that the misclassification probability is allowed to vary across observations depending on the corresponding true response, but not on covariates. *Misclassification not at random* describes the misclassification mechanism that is neither misclassification completely at random nor misclassification at random, for example due to misclassification of responses depending on covariates or random variation. In computer science literature (Frénay and Verleysen, 2013; Manwani and Sastry, 2013; Ghosh et al., 2015) response misclassification is referred to as *label noise* and misclassification completely at random is termed *uniform noise*, while misclassification at random is termed *class conditional noise*.

Alternatively, misclassification can be categorised into two types depending on whether the probability of mislabelling a 1 and the probability of mislabelling a 0 varies across observations. This leads to the terminology *differential* and *non-differential misclassification* (Magder and Hughes, 1997). *Non-differential misclassification* is equivalent to *misclassification at random*, where the probability of misclassification is constant for all observations with the same true response. This includes the case of *misclassification completely at random* where the probability of mislabelling a 1 and the probability of mislabelling a 0 are
the same constant. *Differential misclassification* refers to the case where the probability of an observation being misclassified is not constant across all observations, either due to dependence on covariates or variability in the observation process. Section 1.2.1 gives more formal definitions of misclassification types.

![Figure 1.1: Different types of misclassification according to different taxonomies in Statistics and machine learning literature where the probability of misclassifying the $i$th observation is either $\gamma_{0i} = p(y_i = 1|t_i = 0, x_i)$ or $\gamma_{1i} = p(y_i = 0|t_i = 1, x_i)$; $t_i$ and $y_i$ are the true response and observed response of the $i$th observation, $x_i$ are the observed covariates and $v_i$ are the covariates that determine the misclassification probabilities, which can include $x_i$, but may also contain, for example, additional variability.](image)

The primary difficulty of modelling under misclassification is that, in order to learn misclassification probabilities, we either need external information about misclassification probabilities, validation data where both true and observed responses are available, or additional modelling assumptions. Statistical methods addressing misclassification have appeared in the literature in a variety of contexts, sometimes using different terminology, and often focusing on a particular application (Neuhaus, 1999; Daniel Paulino et al., 2003; Magder and Hughes, 1997; Edwards et al., 2013). This thesis is motivated by the pervasiveness of label noise in a wide variety of applications, coupled with the lack of unified statistical exposition and comparison of all available methods. We first cast existing statistical methods under the same likelihood-based framework and then propose a Dirichlet process probit
misclassification mixture model for flexible modelling of misclassification in binary data in Chapter 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Descriptions</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM</td>
<td>Use expectation maximisation with likelihood including observed responses and unobserved true response.</td>
<td>Magder and Hughes (1997)</td>
</tr>
<tr>
<td>Post-fit adjustment of regression coefficients</td>
<td>Fit a generalised linear model to observed responses, then adjust regression coefficients.</td>
<td>Neuhaus (1999)</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>Simulate more data with added mislabelling to learn error trend.</td>
<td>Küchenhoff et al. (2006)</td>
</tr>
<tr>
<td>Two-step modelling</td>
<td>Use validation group to estimate misclassification probabilities first, then maximise the joint likelihood of validation and main data.</td>
<td>Lyles et al. (2011)</td>
</tr>
<tr>
<td>Full bayesian modelling</td>
<td>Assume prior distribution for parameters of misclassification probabilities, fit joint Bayesian model on both main study group and validation data.</td>
<td>Katz and Katz (2010)</td>
</tr>
<tr>
<td>MI + ML</td>
<td>Use multiple imputation to estimate true responses, then maximise the likelihood of imputed true responses.</td>
<td>Edwards et al. (2013)</td>
</tr>
</tbody>
</table>

Table 1.1: Summary of methods described in detail in this thesis where MEL is maximum estimated likelihood, and ML is maximum likelihood. The first four methods assume known non-differential misclassification probability while the last three methods require validation data whose true responses are known in addition to the observed responses.

1.2 General framework

1.2.1 Notation

We introduce notation which we will use throughout the thesis. Bold symbols are used to denote vectors and capital letters to represent matrices. Data are denoted by $[X, y]$ where $X$ represents the $n \times p$ covariate matrix and $y$ represents the $n \times 1$ observed response vector. The
ith observation is \((x_i, y_i)\) where row vector \(x_i = (x_{i1}, \ldots, x_{ip})\) represents \(p\) covariate values for the \(i\)th observation and \(y_i\) takes in \(\{0, 1\}\). The observed binary responses \(y = (y_1, \ldots, y_n)^T\) are assumed to be a contaminated version of unobserved true responses \(t = (t_1, \ldots, t_n)^T\).

For misclassification probabilities, \(\gamma_{0i}\) is the probability of the \(i\)th observed response being 1 given the corresponding true response is 0 and extra attributes and \(\gamma_{1i}\) is the probability of the observation being 0 given its true response is 1 and extra attributes. Specifically,

\[
\gamma_{0i} = p(y_i = 1 \mid t_i = 0, v_i) \quad \text{and} \quad \gamma_{1i} = p(y_i = 0 \mid t_i = 1, v_i) \quad (i = 1, \ldots, n),
\]

(1.1)

where \(v_i\) represents the attributes related to misclassification probability for the \(i\)th observed response that may or may not overlap with covariates contained in the data \(x_i\). This definition in (1.1) reflects the general setting where the probability of misclassifying a 1 or 0 varies across observations, known as differential misclassification. The case of non-differential misclassification is a special case of Equation (1.1) where

\[
\gamma_{0i} = \gamma_0, \quad \text{and} \quad \gamma_{1i} = \gamma_1 \quad (i = 1, \ldots, n).
\]

Most model-based methods reviewed here are based on Generalised Linear Models (GLMs), in which the components of the true response vector \(t = (t_1, \ldots, t_n)^T\) are assumed to be independently distributed with means \(\mu = (\mu_1, \ldots, \mu_n)^T\) and an exponential family distribution with density or probability mass function of the form

\[
f(t_i; \eta_i, \phi) = \exp\{ (t_i \eta_i - b(\eta_i))/a(\phi) + c(t_i, \phi) \}
\]

for some functions \(a(\cdot), b(\cdot)\) and \(c(\cdot)\) (McCullagh and Nelder, 1989). A GLM links the mean \(\mu_i\) to a linear predictor \(\eta_i = \beta_0 + x_i \beta\) through a monotone, sufficiently smooth link function.
$g(\mu_i) = \eta_i$ where $(\beta_0, \beta_1, \cdots, \beta_p)^T$ is the vector of model regression parameters (McCullagh and Nelder, 1989). In the framework of GLMs, the expected value of the true binary response for the $i$th observed response is modelled as

$$g\{p(t_i = 1 \mid x_i, \beta, \beta_0)\} = \beta_0 + x_i \beta.$$  

### 1.2.2 Likelihood approach

This thesis focuses on likelihood-based methods, where the likelihood of the observed data is

$$L(\beta, \beta_0; x_i, y_i) = \prod_{i=1}^{n} p(y_i = 1 \mid x_i, \beta, \beta_0)^{y_i} [1 - p(y_i = 1 \mid x_i, \beta, \beta_0)]^{1-y_i}. \tag{1.2}$$

Using $\gamma_0$ and $\gamma_1$ to denote misclassification vectors for all observations, i.e $\gamma_0 = (\gamma_{01}, \cdots, \gamma_{0n})^T$ and $\gamma_1 = (\gamma_{11}, \cdots, \gamma_{1n})^T$, when there is no misclassification, the observed responses are the true responses and thus follow the GLM so that $p(y_i = 1 \mid x_i, \beta, \beta_0) = p(t_i = 1 \mid x_i, \beta, \beta_0) = g^{-1}(\beta_0 + x_i \beta)$. When $y_i$ may be misclassified, the probability of observed responses can be written in terms of the probability of true responses and misclassification probabilities where

$$p(y_i = 1 \mid x_i, \beta, \beta_0, \gamma_0, \gamma_1) = (1 - \gamma_{1i} - \gamma_{0i})p(t_i = 1 \mid x_i, \beta, \beta_0) + \gamma_{0i}. \tag{1.3}$$

The challenge with the likelihood based on (1.3) is that the true responses are typically not observed and the misclassification probabilities may not be known. Likelihood-based approaches for misclassification attempt to overcome this challenge in one of the three ways: i) assuming known misclassification probabilities (for example, if the sensitivity and specificity of a medical test are known), ii) relying on a set of available true responses (such as a subset of the data where both the true and observed responses are available), iii) through extra
modelling assumptions.

1.3 Known misclassification probabilities

We first discuss methods that operate under the assumption that misclassification probabilities are known. The methods in this section are introduced under non-differential misclassification, but many methods can be easily extended to the case of differential misclassification. If the misclassification probabilities are non-differential and known, then (1.3) can be simplified to

\[ p(y_i = 1 | x_i, \beta, \beta_0, \gamma_0, \gamma_1) = (1 - \gamma_1 - \gamma_0)p(t_i = 1 | x_i, \beta, \beta_0) + \gamma_0. \]

The aim is to draw inferences about \( \beta, \beta_0 \) accounting for the misclassification probabilities.

1.3.1 Expectation-Maximisation

Magder and Hughes (1997) assumed a logistic regression model for the true responses and proposed an Expectation-Maximisation (EM) algorithm to iteratively estimate the regression coefficients assuming known non-differential misclassification probabilities \( \gamma_0 \) and \( \gamma_1 \). Under this assumption, based on (1.2), the incomplete-data log-likelihood of the observed data can be written as

\[
l(\beta, \beta_0; y, X, \gamma_0, \gamma_1) = \sum_{i=1}^{n} y_i \log \{ p(t_i = 1 | x_i, \beta, \beta_0)(1 - \gamma_1) + p(t_i = 0 | x_i, \beta, \beta_0)\gamma_0 \} + \\
\sum_{i=1}^{n} (1 - y_i) \log \{ p(t_i = 1 | x_i, \beta, \beta_0)\gamma_1 + p(t_i = 0 | x_i, \beta, \beta_0)(1 - \gamma_0) \}.
\]

(1.4)
The complete-data log-likelihood including the true responses \( t_i \) in addition to observed responses \( y_i \) and covariates \( x_i \), is

\[
l(\beta, \beta_0; y, X, t, \gamma_0, \gamma_1) = \sum_{i=1}^{n} \{ t_i \log p(t_i = 1 | x_i, \beta, \beta_0) + (1 - t_i) \log p(t_i = 0 | x_i, \beta, \beta_0) \} \\
+ \sum_{i=1}^{n} \{ t_i [y_i \log(1 - \gamma_1) + (1 - y_i) \log(\gamma_1)] \\
+ (1 - t_i) [(1 - y_i) \log(1 - \gamma_0) + y_i \log(\gamma_0)] \}.
\]

(1.5)

The EM algorithm alternates between the E-step of determining the conditional expectation of (1.5) and the M-step that maximises the conditional expectation of the complete-data log-likelihood with respect to \( \beta \). Since the complete data log-likelihood (1.5) is linear in \( t_i \), the E-step simplifies to replacing \( t_i \) in (1.5) with \( \hat{t}_i = \mathbb{E}(t_i | y_i, x_i, \beta, \beta_0) \). By Bayes’ theorem,

\[
p(t_i = 1 | y_i = 1, x_i, \beta, \beta_0) = \frac{p(t_i = 1, y_i = 1 | x_i, \beta, \beta_0)}{p(t_i = 1, y_i = 1 | x_i, \beta, \beta_0) + p(t_i = 0, y_i = 1 | x_i, \beta, \beta_0)}, \\
p(t_i = 1 | y_i = 0, x_i, \beta, \beta_0) = \frac{p(t_i = 1, y_i = 0 | x_i, \beta, \beta_0)}{p(t_i = 1, y_i = 0 | x_i, \beta, \beta_0) + p(t_i = 0, y_i = 0 | x_i, \beta, \beta_0)}.
\]

Then, under the assumption of non-differential misclassification

\[
\hat{t}_i = \begin{cases} 
\frac{p(t_i = 1 | x_i, \beta, \beta_0)(1 - \gamma_1)}{p(t_i = 1 | x_i, \beta, \beta_0)(1 - \gamma_1) + p(t_i = 0 | x_i, \beta, \beta_0)\gamma_0}, & \text{if } y_i = 1 \\
\frac{p(t_i = 1 | x_i, \beta, \beta_0)\gamma_1}{p(t_i = 1 | x_i, \beta, \beta_0)\gamma_1 + p(t_i = 0 | x_i, \beta, \beta_0)(1 - \gamma_0)}, & \text{if } y_i = 0.
\end{cases}
\]

(1.6)

Magder and Hughes (1997) assume a logit link for the probability of true responses; the steps of their method are summarised in Algorithm 1. The algorithm iterates until the change in parameter estimates \( \beta^{(k)} \) and \( \beta_0^{(k)} \) all become smaller in absolute value than some pre-determined threshold \( \delta > 0 \).
Algorithm 1: EM

*Input:* observations \((x_1, y_1), \ldots, (x_n, y_n)\), non-differential misclassification probabilities \(\gamma_0, \gamma_1\), initial guesses of parameters of likelihood function \(\beta_0^{(0)}, \beta_1^{(0)}, \ldots\), threshold \(\delta > 0\);

*Output:* Estimates for \(\beta, \beta_0\) for logistic regression;

1. Start with the initial guess of the parameters \(\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}\) for logistic regression and use them to evaluate \(\hat{t}_i^{(1)}\)'s by equation (1.6);

2. Get new estimates of regression parameters \(\beta_0^{(1)}, \beta_1^{(1)}, \ldots, \beta_p^{(1)}\) by maximising the expression of equation (1.5) after replacing \(t_i\) with \(\hat{t}_i^{(1)}\);

3. **while algorithm has not converged do**

   4. At iteration \(k + 1\), use the estimates \(\beta_0^{(k)}, \beta_1^{(k)}, \ldots\) from the previous round to calculate the new \(\hat{t}_i^{(k+1)}\)'s by equation (1.6);

   5. Use the new \(\hat{t}_i^{(k+1)}\)'s to get new estimates of \(\beta_0^{(k+1)}, \beta_1^{(k+1)}, \ldots\) by maximising equation (1.5) again after replacing \(t_i\) with \(\hat{t}_i^{(k+1)}\);

   6. Check if the algorithm has met the criteria for convergence by evaluating whether 
      \[
      \max_j \left( \beta_j^{(k+1)} - \beta_j^{(k)} \right) \leq \delta \text{ where } j = \{0, 1, \ldots, p\}.
      \]

   7. **end**

The EM procedure described above assumes known non-differential misclassification and assumes known values of \(\gamma_0\) and \(\gamma_1\). If these two misclassification probabilities are unknown, Magder and Hughes (1997) alternatively suggests estimating them simultaneously with the \(\beta, \beta_0\) through maximum likelihood using the respective proportions

\[
\hat{\gamma}_0 = \frac{\sum_{(y_0=0)} \hat{t}_i}{\sum \hat{t}_i}, \quad (1.7)
\]
\[
\hat{\gamma}_1 = \frac{\sum_{(y_0=1)} (1 - \hat{t}_i)}{\sum (1 - \hat{t}_i)} \quad (1.8)
\]

at each round of the EM algorithm. However, as noted in Magder and Hughes (1997), 26
estimating $\gamma_0, \gamma_1$ simultaneously with $\beta, \beta_0$ may lead to identifiability issues where two or more parameterisations are observationally equivalent (see B.2 for detailed discussion).

### 1.3.2 Post-fit adjustment of regression coefficients

Neuhaus (1999) investigated the effect of fitting the assumed model of the true observations to the imperfectly observed ones. In other words, the premise of Neuhaus (1999) is that $p(y_i = 1 \mid x_i, \beta)$ is incorrectly modelled using a GLM with

$$g\{p(y_i = 1 \mid x_i, \beta, \beta_0)\} = \beta_0^* + x_i \beta^*,$$  \hspace{1cm} (1.9)

when, in fact, the model should have been

$$g\{p(t_i = 1 \mid x_i, \beta, \beta_0)\} = \beta_0 + x_i \beta_0.$$  \hspace{1cm} (1.10)

Neuhaus (1999) studied the magnitude of the bias caused by using estimators $\beta^*$ for parameters $\beta$ where $\beta^*$ are obtained by modelling the observed responses $y_i$ using a misspecified link function and $\beta_0$ are the true regression coefficients modelling true responses $t_i$ using the correct link function. The post-fit adjustment of regression coefficients proposed by Neuhaus (1999) is achieved by rescaling $\beta^*$ by the magnitude of bias. For general cases, the magnitude of bias can be acquired from the Kullback-Leibler divergence between the true model with (1.10) and misspecified one with (1.9) (Neuhaus, 1999). In the case of known non-differential misclassification probabilities, the magnitude of bias can be approximately computed. The approximation is done element-wise with

$$\beta_j^* = g\left(p(y_i = 1 \mid x_i^+; \beta, \beta_0)\right) - g\left(p(y_i = 1 \mid x_i, \beta, \beta_0)\right) = H_j(\beta),$$  \hspace{1cm} (1.11)
where $x_i^{+j} = (x_{i1}, \ldots, x_{ij} + 1, \ldots, x_{ip})^T$. A Taylor expansion for (1.11) at $\beta = 0$ ignoring terms higher than first order gives

$$\beta_j^* \approx H_j(0) + \beta_j H'_j(0). \quad (1.12)$$

After some algebra (see Appendix A.1), it can be shown that $H_j(0) = 0$ and

$$H'_j(0) = \frac{(1 - \gamma_1 - \gamma_0)g'(1 - \gamma_1 - \gamma_0)g^{-1}(\beta_0) + \gamma_0}{g'[g^{-1}(\beta_0)]} \quad (1.13)$$

where $g'$ is the derivative of link function $g$. The link function $g$ is assumed to be strictly monotone and differentiable (Neuhaus, 1999). Expression (1.13) expresses the (multiplicative) magnitude of bias for all regression coefficients where $\beta_j^* \approx \beta_j H'_j(0)$ for $j \in \{1, \ldots, p\}$.

The magnitude of bias indicates the scale of stretch or compression of the correct coefficients $\beta$ of the true model compared to the $\beta^*$ from the misspecified model. Note that the magnitude of bias is a function of the link function of choice $g$: for logit link, (1.13) simplifies to

$$H'_j(0)_{\text{logit}} = \frac{(1 - \gamma_1 - \gamma_0)e^{\beta_0}}{e^{\beta_0}(1 - \gamma_1) + \gamma_0}\frac{1 - \gamma_1 - \gamma_0}{\gamma_1 e^{\beta_0} + 1 - \gamma_0} \quad (1.14)$$

(see Appendix A.1). Although the goal of Neuhaus (1999) was not to provide a method for post-hoc adjustment of regression coefficients, but rather to investigate the effect of using the misspecified model, the resulting magnitude of bias can be used for coefficient adjustment assuming non-differential misclassification. We summarise the steps of Neuhaus (1999) with coefficient adjustment in Algorithm 2.
Algorithm 2: Post-fit adjustment of regression coefficients

*Input:* observations \((x_1, y_1), \ldots, (x_n, y_n)\), choice of GLM model with strictly monotone and differentiable link function \(g\), known intercept \(\beta_0\), non-differential misclassification probabilities \(\gamma_0, \gamma_1\);

*Output:* Estimates for \(\beta\) for GLM;

1. Fit a GLM to the observed dataset to get \(\beta^*_0, \beta^*\) for misspecified model where \(g\{p(y_i = 1 \mid x_i, \beta, \beta_0)\} = \beta^*_0 + x_i\beta^*\);
2. Calculate the magnitude of bias by plugging \(\beta_0, \gamma_0, \gamma_1\) in equation (1.13);
3. Divide \(\hat{\beta}^*\) by the magnitude of bias to get \(\hat{\beta}\).

The post-fit adjustment of regression coefficients of Neuhaus (1999) has the obvious advantage of simplicity and computational efficiency. However, it relies on the Taylor expansion in (1.12), which means that the approximation is only good when the true \(\beta\) is close to 0. Furthermore, the magnitude of bias \(H'(0)\) in (1.13) is a function of known \(\gamma_0, \gamma_1\) and \(\beta_0\).

1.3.3 Misclassification Simulation Extrapolation

Küchenhoff et al. (2006) proposed a method called the Misclassification Simulation Extrapolation (MC-SIMEX). MC-SIMEX assumes known non-differential misclassification and is an extension of the Simulation Extrapolation (SIMEX) method by Cook and Stefanski (1994) developed for handling measurement error in statistical modelling. MC-SIMEX is intended for measurement error in categorical variables, thus can be applied to misclassification in binary responses.

MC-SIMEX is similar to Neuhaus (1999) in the sense that it relies on fitting the model for the true responses to the contaminated observations, and relating the estimated coefficients to the coefficients that would have been obtained using the true data. As opposed to Neuhaus (1999), where a Taylor expansion is used to obtain the relationship between the
two sets of coefficients, the main idea of MC-SIMEX is to learn the relationship between the level of misclassification and the bias of estimated coefficients by simulating additional data with various levels of misclassification. A parametric function is fitted to model the effect of misclassification on the coefficient estimates, and extrapolation is then used to obtain estimates of the model parameters for data with no misclassification.

MC-SIMEX starts by assuming a model, typically a GLM, for the true responses, and a known misclassification matrix $\Pi$ with the true misclassification probabilities on the off diagonals where

$$
\Pi = \begin{bmatrix}
1 - \gamma_0 & \gamma_0 \\
\gamma_1 & 1 - \gamma_1
\end{bmatrix}.
$$

In the first step, multiple datasets with predetermined misclassification matrix $\Pi'$ are generated by injecting additional noise $\Pi'$ to the (already noisy) data. For each dataset with a specific level of misclassification, a set of estimated model parameters, called naive estimators, are obtained by fitting the model for the true responses to the contaminated ones. Let $\mathcal{G}(\Pi')$ denote the naive estimated model parameters for a dataset simulated with misclassification matrix $\Pi'$. Then $\mathcal{G}(I_2)$, where $I_2$ is the 2-by-2 identity matrix, represents the model parameters when there is no misclassification and is thus referred to as the true estimator. In the second step, after obtaining different sets of $(\Pi', \mathcal{G}(\Pi'))$ for various misclassification matrices $\Pi'$, a parametric model of $\mathcal{G}(\Pi')$ given $\Pi'$ can be used to estimate $\mathcal{G}(I_2)$.

The main challenge in MC-SIMEX is to add misclassification such that all the misclassification matrices are related in a way that allows a parametric model to be used to extrapolate the model parameters back to the case where the misclassification matrix is $I_2$. Küchenhoff et al. (2006) resolves this difficulty by defining $\Pi'$ through an increasing sequence $\lambda_1, \ldots, \lambda_K$ of positive powers of $\Pi$, the known misclassification matrix of the observed data at hand. The
synthetic data at level $k$ have injected misclassification $\Pi' = \Pi^k$, thus total misclassification at $\Pi'\Pi = \Pi^{k+1}$. The proof of why $\Pi^k$ for $\lambda > 0$ corresponds to higher misclassification relies on the eigen-decomposition of $\Pi$ and the assumption that $\gamma_0 + \gamma_1 < 1$. More details, along with a detailed description of how misclassification is added in synthetic datasets is given in Appendix A.2.

For $\lambda = -1$, $\Pi^{k+1} = I_2$, implying no misclassification. Once a series of naive estimators $(\hat{\beta}_0^{(1)}, \beta^{(1)}), \ldots, (\hat{\beta}_0^{(K)}, \beta^{(K)})$ has been obtained, any flexible regression method can be used to model $\mathcal{G}$ as a function of $\lambda$ in order to extrapolate the true estimates of $\beta, \beta_0$, corresponding to $\lambda = -1$.

In practice, to reduce sampling variability, the generation of new datasets is repeated $B$ times for each of $\lambda_1, \ldots, \lambda_K$. Let $MC[\Pi^k](\cdot)$ denote the process of generating a response vector after adding $\Pi^k$ amount of misclassification ($k \in \{1, \ldots, K\}$), then the $B$ sets of new responses with a fixed amount of added misclassification can be expressed as

$$y^{(k,b)} = MC[\Pi^k](y), \quad b = 1, \ldots, B, \quad k = 1, \ldots, K.$$ 

Each $y^{(k,b)}$ is used to obtain estimates $\beta^{(k,b)}, \beta_0^{(k,b)}$. Küchenhoff et al. (2006) use the averages $\bar{\beta}^{(1)}, \ldots, \bar{\beta}^{(K)}$ with $\bar{\beta} = \sum_{b=1}^{B} \beta^{(k,b)}/B$ and $\bar{\beta}_0^{(1)}, \ldots, \bar{\beta}_0^{(K)}$ with $\bar{\beta}_0 = \sum_{b=1}^{B} \beta_0^{(k,b)}/B$ to estimate the regression of $\mathcal{G}$ against $\lambda$ and estimate $\mathcal{G}(I_2)$.

The steps of MC-SIMEX are summarised in Algorithm 3.
Algorithm 3: MC-SIMEX

Input: observations \((x_1, y_1), \ldots, (x_n, y_n)\), choice of parametric model \(M\)
parameterised by \(\beta, \beta_0\), non-differential misclassification probabilities \(\gamma_0, \gamma_1\),
number of datasets to simulate: \(B\), a positive increasing vector \(\lambda = (\lambda_1, \ldots, \lambda_K)\) to
denote added misclassification schedule, choice of parametric extrapolation model;

Output: Estimates for \(\beta, \beta_0\);

1. Fit the model \(M\) to the observed data \(D\) to obtain the naive estimates \(\beta^{(0)}, \beta_0^{(0)}\);
2. Compute the misclassification matrix using (1.15);
3. for \(k = 1:K\) do
   4. for \(b = 1:B\) do
      5. Simulate a new dataset where \(D^{(k,b)} = [X, y^{(k,b)}]\) and \(y^{(k,b)} = MC[\Pi^{\lambda_k}](y)\);
      6. Refit the model \(M\) to the synthetic data \(D^{(k,b)}\) to obtain new parameters \(\beta^{(k,b)}\);
   7. end
   8. Set \(\beta^{(k)} = \sum_{b=1}^B \beta^{(k,b)}/B\) and \(\beta_0^{(k)} = \sum_{b=1}^B \beta_0^{(k,b)}/B\);
9. end
10. Use \(\beta^{(0)}, \beta^{(1)}, \ldots, \beta^{(K)}\) and \(\beta_0^{(0)}, \ldots, \beta_0^{(K)}\) that correspond to \(\lambda = 0, \lambda_1, \ldots, \lambda_K\) to
    estimate parameters in the parametric function and then plug in \(\lambda = -1\) to obtain
    an estimator for \(\beta, \beta_0\).

Like Neuhaus (1999), the MC-SIMEX method of Küchenhoff et al. (2006) takes a similar
approach of relating coefficient estimates of the model fitted to the contaminated data to the
true coefficients of the same model fitted to the true data. However, MC-SIMEX is a more
general method because it tries to learn the relationship directly. It can be used for any
variable with categorical measurement error (i.e., both covariates and response variables)
and can inherently deal with more than two classes in the error-prone variable. On the
other hand, MC-SIMEX is computationally costly as the simulation step is repeated many

32
times for each level of misclassification. The estimated model parameters for data with no misclassification also depend on the extrapolation function, which builds on the assumption that the function $G(\Pi^{\lambda+1})$ can be approximated by fitting a chosen parametric model to pairs of $(\lambda, G(\Pi^{\lambda}))$. Note that the extrapolation also implicitly requires the relationship between the parameters $\beta$ and misclassification matrix to be the same when the level of misclassification increase, for which there is no guarantee.

1.3.4 Maximum estimated likelihood

Motivated by the need to deal with both contaminated observed responses and unidentifiable model parameters in the case of complete separation in the covariates space, Rousseeuw and Christmann (2003) propose the hidden logistic regression model. The main idea of Rousseeuw and Christmann (2003) is to maximise a likelihood similar to (1.3) but replacing each $y_i$ with pseudo-observation $\tilde{y}_i$. This likelihood based on the pseudo-observations is an approximated likelihood and the method is called the maximum estimated likelihood.

Rousseeuw and Christmann (2003) assume known non-differential misclassification probabilities, and further assume $\gamma_0 < 0.5$ and $\gamma_1 < 0.5$. This implies that, when $t_i = 1$, the likelihood of $y_i = 1$ exceeds that of $y_i = 0$, and vice versa when $t_i = 0$. Therefore, the maximum likelihood estimator of $t_i$ is $\hat{t}_i = y_i$. The pseudo-observation $\tilde{y}_i$ is defined as the expectation of the $i$th observed response conditional on the maximum likelihood estimate of $t_i$

\[
\tilde{y}_i = p(y_i = 1|\hat{t}_i = 0)p(\hat{t}_i = 0) + p(y_i = 1|\hat{t}_i = 1)p(\hat{t}_i = 1) = \gamma_0(1 - y_i) + (1 - \gamma_1)y_i.
\]

If the response is 1, then the maximum likelihood estimate of true response is 1 and vice versa. This leads to the maximum estimated likelihood (MEL), whose log likelihood can be
written as

\[
\sum_{i=1}^{n} \tilde{y}_i \log[g^{-1}(x_i\beta + \beta_0)] + (1 - \tilde{y}_i) \log[1 - g^{-1}(x_i\beta + \beta_0)].
\] (1.17)

The estimator that maximises (1.17) is the MEL estimator, denoted by \( \hat{\beta}_{\text{MEL}} \) (Rousseeuw and Christmann, 2003). A nice property of the MEL estimator is that it always exists and is unique. The explanation is given in Appendix A.3.

To increase MEL estimator’s robustness to outliers, Rousseeuw and Christmann (2003) further proposed the *weighted maximum estimated likelihood estimator*. The reasoning follows from the observation that in the score function of MEL (see (A.1) in Appendix A.3), the impact of bad leverage points is unbounded, thus the MEL estimator could be heavily affected by outliers. Rousseeuw and Christmann (2003) proposed to downweight bad leverage points by adding ‘bounded’ weights \( w_i \) to the likelihood, leading to new objective of solving

\[
\prod_{i=1}^{n} [\tilde{y}_i - g^{-1}(x_i\beta + \beta_0)]w_i x_i = 0.
\] (1.18)

The solution of (1.18) is referred to as the *weighted maximum estimated likelihood* (WEMEL) (Rousseeuw and Christmann, 2003). The *weight* for the \( i \)th observation, \( w_i \), is bounded between 0 and 1 and is defined as

\[
w_i = \frac{M}{\max\{RD^2(x_i^*), M\}}
\] (1.19)

where \( RD(x_i^*) \) stands for the robust distance (of an observation \( x_i \)) and \( M \) is the 75th percentile of all \( RD^2(x_i^*) \). The implication of equation (1.18) is that roughly 75\% of design points have weights of 1 and the 25\% of the most extreme design points have weights less than 1. Details about robust distance is given in Appendix A.3.
The steps for MEL and WEMEL are summarised in Algorithm 4. Note that the implementation procedure of WEMEL is very similar to that of MEL, with the addition of weight \( w_i \) used in the likelihood function, which only needs to be calculated once before the optimisation process. Although the methods in Rousseeuw and Christmann (2003) were developed for non-differential misclassification probabilities, they can also be applied on known differential misclassification settings provided \( \gamma_0 < 0.5 \) and \( \gamma_1 < 0.5 \) for all \( i \).

**Algorithm 4: MEL (and WEMEL)**

*Input:* observations \((x_1, y_1), \ldots, (x_n, y_n)\), choice of parametric model \( \mathcal{M} \) parameterised by \( \beta, \beta_0 \), non-differential misclassification probabilities \( \gamma_0, \gamma_1 \);

*Output:* Estimates for \( \beta, \beta_0 \);

1. (for WEMEL, calculate the robust distance for all samples, \( RD(x_i) \));
2. Compute pseudo-observations \( \tilde{y}_i = (1 - y_i)\gamma_0 + y_i(1 - \gamma_1) \);
3. Compute weights \( w_i \) (for MEL, \( w_i = 1 \) for all \( i \));
4. Solve equation equation (1.18) to get estimates for \( \beta, \beta_0 \).

The MEL estimator of Rousseeuw and Christmann (2003) obtained by maximising a modified likelihood function is simple and fast to implement. It also has nice properties like guaranteed existence and uniqueness of estimators. The WEMEL estimator, which is intended to be more robust against outliers, requires calculation of weights which depend on robust distances and are computationally expensive. Both methods are intended for use when the level of misclassification is low (Rousseeuw and Christmann, 2003).

### 1.4 Modelling unknown misclassification probabilities by assuming validation data

The methods that we have described so far all assume known misclassification probabilities. In practice, the misclassification probabilities might not be known. In this section,
we review methods that replace the known misclassification probabilities assumption with other assumptions, for example the availability of validation data where true responses of observations are also available.

One popular approach to estimate misclassification probabilities is assuming the availability of validation data where both observed and true responses are given. In this approach, the data at hand are assumed to be divided into two groups: the main study group $S$ and the validation group $V$. In the main study group, the covariates $x_i$ and observed responses $y_i$ are available. In the validation group, the true responses $t_i$ are also observed, in addition to $x_i$ and $y_i$. The key underlying assumption is that the relationship of true responses on observed responses and covariates is the same for $V$ and $S$.

The likelihood from the main study group $L_S$ is unchanged and follows straight from (1.2) and (1.3). The likelihood from the validation group $L_V$ needs to account for the four possible combinations of $(y_i, t_i)$, leading to

$$L_V = \prod_{i \in V} p(y_i = 1, t_i = 1 | x_i)^{y_it_i} p(y_i = 0, t_i = 1 | x_i)^{(1-y_it_i)} \times p(y_i = 1, t_i = 0 | x_i)^{y_i(1-t_i)} p(y_i = 0, t_i = 0 | x_i)^{(1-y_i)(1-t_i)}.$$  

The joint likelihood of data is then given by

$$L_{S,V} = L_S \times L_V.$$  

1.4.1 Two-step modelling

The method of Lyles et al. (2011) assumes differential misclassification and can be broken down into two steps. The first step uses the validation data to learn misclassification probabilities. The second step uses learned misclassification probabilities to infer the parameters.
of interest $\beta$ by maximising the overall likelihood composed of the product of $L_V$ and $L_S$. Specifically, the misclassification probabilities are assumed to be a deterministic function of the covariates where we can write

$$
\eta_{it} = g(p(y_i = 1 \mid t_i = t, x_i)) = \alpha_0 + \alpha_1 t + \sum_{k=1}^{p} \alpha_{k+1} x_{ik}
$$

(1.22)

with $t \in \{0, 1\}$ (Lyles et al., 2011). Given the values of the covariates and the true response, the probability of the observed response being 1 is fixed. The incorporation of $x_{i1}, \ldots, x_{ip}$ in $\eta_{it}$ implies differential misclassification. If $x_{i1}, \ldots, x_{ip}$ are omitted in (1.22) or $\alpha_2, \ldots, \alpha_{p+1}$ are set to zero, then the misclassification probabilities are non-differential. The $g(\cdot)$ in (1.22) can, in principle, be any link function but a logit link is used Lyles et al. (2011). Fitting the observations in validation data $V$ with (1.22) provides estimates for $\hat{\alpha}_0, \hat{\alpha}_1, \ldots, \hat{\alpha}_{p+1}$ which can then be used to estimate $\hat{\gamma}_{0i}, \hat{\gamma}_{1i}$ for any observation $i$ by plugging in $t = 0$ and $t = 1$, including those in main study group $S$. This implicitly assumes observations in $V$ and $S$ have the same misclassification mechanism and estimates for misclassification probabilities $\gamma_{0i}, \gamma_{1i}$ can be obtained through

$$
\hat{\gamma}_{1i} = 1 - g^{-1}(\hat{\eta}_{1i}) \quad \text{and} \quad \hat{\gamma}_{0i} = g^{-1}(\hat{\eta}_{0i}).
$$

(1.23)

Since $L_S$ and $L_V$ are both functions of $\gamma_{0i}, \gamma_{1i}$ and $\beta, \beta_0$, once the misclassification probabilities are estimated, the only unknowns in $L_S$ and $L_V$ are $\beta, \beta_0$ that parametrise the probabilities of true responses. Therefore, maximising the final (overall) likelihood $L = L_V \times L_S$ provides estimates for $\beta, \beta_0$. All steps of Lyles et al. (2011) are summarised in Algorithm 5.
Algorithm 5: Two-step modelling

**Input:** data \(\{S, V\}\) where \(S\) represents the main study group in which we observe \(x_i, y_i \in S\) and \(V\) is the validation group in which we observe \(x_i, y_i, t_i \in V\) where \(t_i\) is the true response for observation \(i\), choice of parametric model \(\mathcal{M}\) parameterised by \(\beta, \beta_0\);

**Output:** Estimates for \(\beta, \beta_0\);

1. **Step one:** Fit a logistic regression regressing \(y\) on \(t\) and \(X\) using observations in the validation data \(V\) through (1.22) where \(\eta_{it} = \alpha_0 + \alpha_1 t + \sum_{k=1}^{p} \alpha_{k+1} x_{ik}\);

Use the \(\alpha_j\)'s and (1.22) and (1.23) to estimate \(\gamma_{0i}, \gamma_{1i}\) for \(i \in V\) and \(i \in S\);

2. **Step two:** Use \(\hat{\gamma}_{0i}, \hat{\gamma}_{1i}\) to maximise the overall joint likelihood \(L = L_V \times L_S\) to get estimates for \(\beta, \beta_0\).

### 1.4.2 Full Bayesian parametric modelling

Katz and Katz (2010) could be viewed as the Bayesian version of Lyles et al. (2011) since the same likelihoods of \(L_S\) and \(L_V\) are used but modelled in a Bayesian manner. Using \(\gamma_{0i}, \gamma_{1i}\) for misclassification probabilities of the \(i\)th observation, the expression for misclassification probabilities in Katz and Katz (2010) are written as

\[
\gamma_{0i} = f(z_{0i}^T \psi_0), \quad \gamma_{1i} = f(z_{0i}^T \psi_1),
\]

where \(z_{0i}, z_{1i}\) are the relevant covariates for modelling \(\gamma_{0i}, \gamma_{1i}\) and \(\psi_0, \psi_1\) their corresponding model coefficients. The true responses are still assumed to follow a GLM where

\[
p(t_i = 1 | x_i, \beta) = g^{-1}(x_i \beta + \beta_0).
\]
Katz and Katz (2010) assume prior distributions for the unknown parameters $\psi_0, \psi_1, \beta, \beta_0$ and model misclassification probabilities and GLM model parameters jointly through

$$p(\beta, \beta_0, \gamma_0, \gamma_1 | \mathcal{S}) \propto \mathcal{L}(\beta, \beta_0, \gamma_0, \gamma_1 | \mathcal{S}) \times \mathcal{L}(\beta, \beta_0, \psi_0, \psi_1 | \mathcal{V})^\zeta \times p(\beta_0) \times p(\beta) \times p(\psi_0) \times p(\psi_1)$$

(1.24)

where prior restrictions are put on $\psi_0, \psi_1$ to constrain misclassification probabilities and $\beta, \beta_0$. The tuning parameter $\zeta$ in (1.24) controls how much influence the likelihood of the validation data has on the main study data and is bounded between 0 and 1 with $\zeta = 0$ corresponding to the case that no auxiliary information is incorporated from $\mathcal{V}$ and $\zeta = 1$ corresponding to the case that equal weights are given to observations of $\mathcal{V}$ and $\mathcal{S}$. Alternatively, a distribution can be placed on $\zeta$ instead of using a fixed value.

In Katz and Katz (2010), the probability of $t_i = 1$ and misclassification probabilities are all assumed to follow a GLM with probit link where

$$p(t_i = 1 | x_i, \beta, \beta_0) = \Phi(x_i/\beta + \beta_0), \quad \gamma_{0i} = \Phi(z_{0i}^T \psi_0), \quad \gamma_{1i} = \Phi(z_{1i}^T \psi_1).$$

Consequently, following the decomposition in equation (1.3), the likelihood functions for the main study and validation data take the same form as in Lyles et al. (2011), namely

$$L_S(\beta, \beta_0, \gamma_{01}, \ldots, \gamma_{0n}, \gamma_{11}, \ldots, \gamma_{1n} | \mathcal{S}) = \prod_{i \in \mathcal{S}} \left\{ [1 - \gamma_{1i} - \gamma_{0i}] \Phi(x_i/\beta + \beta_0) + \gamma_{0i} \right\}^{y_i}$$

$$\times \{ [1 - \gamma_{1i} - \gamma_{0i}] [1 - \Phi(x_i/\beta + \beta_0)] + \gamma_{1i} \}^{1-y_i}$$

(1.25)
and

\[L_V(\beta, \beta_0, \psi_0, \psi_1 | \mathcal{V}) = \prod_{i \in \mathcal{V}} \Phi(x_i \beta + \beta_0)^{t_i} \left[ 1 - \Phi(x_i \beta + \beta_0) \right]^{1 - t_i} \times \prod_{z_i = 0} \Phi(z_i^T \psi_0)^{y_i} \left[ 1 - \Phi(z_i^T \psi_0) \right]^{1 - y_i} \times \prod_{z_i = 1} \Phi(z_i^T \psi_1)^{1 - y_i} \left[ 1 - \Phi(z_i^T \psi_1) \right]^{y_i}.\]  

(1.26)

Applied to a dataset about voter turnout, Katz and Katz (2010) further assume normal priors for \( \psi_0, \psi_1 \) and \( \beta, \beta_0 \). As a result, this prior assumption leads to a conjugate model where full conditionals in (1.24) are in closed-form and can be sampled via a Gibbs sampler.

### 1.4.3 Multiple imputation and maximum likelihood

Edwards et al. (2013) use the validation data to inform the contaminated responses of the main study group data through the use of multiple imputation, accounting for the unobserved true responses and their inherent uncertainty (Little and Rubin, 2014). Similarly to EM of Magder and Hughes (1997), Edwards et al. (2013) view the true responses \( t_i \) in the main study group as missing and treat response misclassification as a missing data problem. Edwards et al. (2013) first impute \( t_i \) for \( i \in \mathcal{S} \) based on observations in the validation data \( \mathcal{V} \), where \( y_i, t_i, x_i \) are all observed, and then build a model for \( t_i | x_i \) based on \( x_i, t_i \) from the complete data \( (\mathcal{S}, \mathcal{V}) \). We refer to this method as MI + ML. To impute the true responses \( t_i \) for \( i \in \mathcal{S} \), a logistic regression of \( t_i \) on \( y_i \) and \( x_i \) is trained on validation data such that

\[g(p(t_i = 1 | y_i, x_i)) = \alpha_0 + x_i \alpha_1 + \alpha_2 y_i.\]  

(1.27)

This step of modelling \( t_i \) based on \( x_i, y_i \), in some sense, is the reverse of Lyles et al. (2011), where \( y_i \) was modelled based on \( x_i, t_i \). Once the relationship between \( t_i \) and \( y_i, x_i \) is learned through the validation data, the observed \( x_i \) and \( y_i \) of the main study data are used to
create $K$ realisations of true responses $t_i$ for $i \in S$ based on (1.27). The version of multiple imputation used by Edwards et al. (2013) assumes that the imputed true response for the $i$th observation in the $k$th imputed dataset, $t_i^{(k)}$ is generated through

$$t_i^{(k)} \sim Bern \left( g^{-1} \left( \hat{\alpha}_0^{(k)} + x_i \hat{\alpha}_1^{(k)} + \hat{\alpha}_2^{(k)} y_i \right) \right)$$  \hspace{1cm} (1.28)$$

where $\hat{\alpha}_0^{(k)}, \hat{\alpha}_1^{(k)}, \hat{\alpha}_2^{(k)}$ are drawn from a multivariate normal distribution centred at the maximum likelihood estimates and with variance-covariance matrix the inverse of the Fisher information at the maximum likelihood estimates.

The $K$ imputed datasets can then each be analysed separately and pooled together using Rubin’s rules (see Appendix A.4). This takes into account of the sampling variability from both sample data and missing values (Little and Rubin, 2014). We summarise the steps of Edwards et al. (2013) in Algorithm 6.

41
Algorithm 6: MI + ML

*Input:* data \{\mathcal{S}, \mathcal{V}\} where \mathcal{S} represents the main study group in which we observe \(x_i, y_i \in \mathcal{S}\) and \mathcal{V} is the validation group in which we observe \(x_i, y_i, t_i \in \mathcal{V}\) where \(t_i\) is the true response for observation \(i\), choice of parametric model \(\mathcal{M}\) parameterised by \(\mathbf{\beta}, \beta_0\);

*Output:* Estimates for \(\mathbf{\beta}, \beta_0\) with variance estimated using Rubin’s rule;

1. Fit a logistic regression regressing \(t\) on \(y\) and \(X\) using \(j \in \mathcal{V}\) through equation (1.27);
2. for \(k=1:K\) do
   3. Compute Gaussian approximation of \(\tilde{\alpha}_0, \tilde{\alpha}_1, \tilde{\alpha}_2\) with mean vector \((\hat{\alpha}_0, \hat{\alpha}_1, \hat{\alpha}_2)\) and covariance matrix estimated from (1.27);
   4. *Impute:* create a new dataset \(D^{(k)} = [X, t^{(k)}]\) where \(t^{(k)} = t_1^{(k)}, \ldots, t_n^{(k)}\) with \(t_i^{(k)} = t_i\) (which is known) for \(i \in \mathcal{V}\) and \(t_i^{(k)} \sim \text{Bern} \left( g^{-1} \left( \tilde{\alpha}_0^{(k)} + x_i \tilde{\alpha}_1^{(k)} + \tilde{\alpha}_2^{(k)} y_i \right) \right)\) for \(i \in \mathcal{S}\) where \(\tilde{\alpha}_0^{(k)}, \tilde{\alpha}_1^{(k)}, \tilde{\alpha}_2^{(k)}\) are drawn from Gaussian approximation of \(\alpha_0, \alpha_1, \alpha_2\) based on (1.27);
   5. *Analyze:* fit model \(\mathcal{M}\) to each imputed dataset \(D^{(k)}\) to obtain estimates \(\hat{\beta}^{(k)}, \hat{\beta}_0^{(k)}\) that parameterise \(\mathcal{M}\);
3. end

7. *Pool:* Follow Rubin’s rule to calculate the mean and variance of our estimator \(\hat{\mathbf{\beta}}, \hat{\beta}_0\) from estimates \(\beta^{(1)}, \ldots, \beta^{(K)}\) and \(\beta_0^{(1)}, \ldots, \beta_0^{(K)}\).

Note that the initial motivation of MI+ML of Edwards et al. (2013) was a case-control study and the main interest was the odds ratio estimates. Nevertheless, the MI+ML approach can be applied when the inference interest is regression coefficients of a pre-specified model. Drawing \(\tilde{\alpha}_0^{(k)}, \tilde{\alpha}_1^{(k)}, \tilde{\alpha}_2^{(k)}\) from the Gaussian approximation of \(\alpha_0, \alpha_1, \alpha_2\) to simulate \(t_i^{(k)}\) is effectively doing Bayesian multiple imputation.
1.5 Other methods

Besides the methods we have discussed so far for modelling mislabelled binary data, there are other methods that make different model assumptions or aim to deal with misclassification through different means.

A somewhat relevant model that also uses EM algorithm like Magder and Hughes (1997) is the stochastic supervision model of Titterington (1989). Assuming we observe assessments of observations, which are probabilities of responses, Titterington (1989) propose an EM algorithm for discriminant analysis that further assumes that the assessments conditioning on true responses are Beta distributed and the covariates conditioning on true response are generated from different Gaussian distributions with the same covariance. For binary data, Lu et al. (2021) generalise the stochastic model to account for the asymmetry case of different covariance matrix of the Gaussian that generate the covariates and allow feature-dependent assessments.

1.5.1 Bayesian modelling using a prior on misclassification probabilities

In cases where validation data are not available, prior distributions on the misclassification probabilities allow the model for the true responses and the misclassification probabilities to be learnt jointly. Beta priors are a common choice of prior for $\gamma_{i0}$ and $\gamma_{i1}$ (Joseph et al., 1995; Gaba, 1993; Rekaya et al., 2001; Zhang et al., 2005; Johnson and Gastwirth, 1991; Joseph and Gyorkos, 1996; Gustafson et al., 2001; Daniel Paulino et al., 2003).

Existing methods have used a variety of simplifications to the general framework. Johnson and Gastwirth (1991) used a Bayesian model without covariates, i.e., non-differential misclassification and a fixed probability in the true data, and placed Beta priors on $p(t_i = 1)$,
\( \gamma_0 \) and \( \gamma_1 \). In Gaba (1993), one of the misclassification probabilities is assumed to be 0 while the other is assumed to follow a Beta prior. In the same context, Hahn et al. (2016) assume only false negatives and a modular prior is placed on the other misclassification probability. Besides Beta priors for misclassification probabilities, its \( k \)-variate generalisation, Dirichlet priors have also been used in some approaches (see Ruiz et al., 2008; Liu et al., 2009). In Ruiz et al. (2008), the interest of inference is the proportions of positives, a multinomial model is assumed and Dirichlet prior is put on misclassification probabilities. In Liu et al. (2009), the inference interest is the odds ratio of case-control study where both exposure level and disease status are potentially misclassified. Dirichlet prior is put on misclassification probabilities of exposure status and Beta prior is put on sensitivity and specificity of disease outcome. In these Bayesian approaches, posterior distributions of parameters are generally difficult to calculate directly, but MCMC methods can be used to approximate posterior of model parameters.

### 1.5.2 Non model-based approaches

This thesis focuses on methods which explicitly model the misclassification process in the likelihood. However, there are many other relevant methods, which we briefly review here. These methods aim to either obtain a ‘noise-tolerant’ (mislabelling-tolerant) classifier, or to **denoise** the data before model building. Denoising effectively means detecting potentially mislabelled (noisy) instances and then modifying the data accordingly. Modifying the data typically implies changing the labels of instances that are detected as potentially mislabelled or eliminating them (Pechenizkiy et al., 2006).

The identification of potentially mislabelled instances typically relies on implicit modelling assumptions, largely relating to a smooth response surface as a function of covariates. Their detection can be implemented by prediction-based models or neighbour-based meth-
ods. Prediction-based models mainly revolve around building one or more classifiers and then utilising their prediction results. Usually, some manipulation of the data (like resampling and subsetting) is used, along with ensemble methods. Neighbour-based methods (or local learning methods) mainly assume misclassified instances have labels which disagree with their neighbors. For instance, Badenas et al. (2000) and Sánchez et al. (2003) detect mislabelled instances through measuring the inconsistency between an instance and its neighbours and employ an iterative procedure that removes suspicious instances on each round.

Generally speaking, *denoising* or pre-processing the data is cheap and easy, but it typically comes at the cost of removing a substantial amount of data or reducing the quality of data (Frénay and Verleysen, 2013). On the other hand, building so-called ‘noise-tolerant models’ frequently comes down to reducing the impact of potentially mislabelled instances in a specific model and is often achieved by avoiding overfitting through regularisation (Teng, 2000, 2001). In some other works, misclassification is studied under the framework of *empirical loss minimisation* (ERM) in which the goal of model building and learning is to find a classifier that minimises a risk function, defined as the expected value of a chosen loss function. For example, the Bayes classifier minimises the risk with respect to 0-1 loss and soft margin SVM minimises hinge loss (Friedman et al., 2001). Under ERM, the goal is to find a modified loss function such that the probability of misclassification under noisy data is the same as under minimising the loss function for mislabelling-free data (Manwani and Sastry, 2013; Ghosh et al., 2015). Constructing the modified loss function requires knowing the misclassification probabilities which are also assumed to be non-differential.

### 1.5.3 Summary

So far we have explained a wide variety of methods for analysing binary data under the presence of response misclassification. We described methods for modelling misclassified binary
data in groups depending on whether they treat misclassification probabilities as known or unknown. Assuming known misclassification, one can use the method of Neuhaus (1999), which approximates the bias of an assumed model through Taylor series and subsequently use it to adjust the coefficients of the model directly fitted to observed data, or the method of Rousseeuw and Christmann (2003) that maximises an alternative likelihood incorporating the misclassification probabilities. On the other hand, it is an option to use EM to iteratively estimate parameters of a specified model as done in (Magder and Hughes, 1997) or simulate more data with different amount of misclassification to learn the effects of misclassification on model parameters for a pre-specified model and therefore attain estimates of the model parameters for that pre-specified model in the absence of misclassification following the proposal in Küchenhoff et al. (2006). When the misclassification probabilities are known, distributional assumptions about the misclassification probabilities are needed or validation data which can be used to learn the misclassification probabilities. Lyles et al. (2011) uses the validation data to learn misclassification probabilities and Edwards et al. (2013) uses the validation data to impute true responses, which are treated as missing. Katz and Katz (2010) resembles the Bayesian version of (Lyles et al., 2011) in which the misclassification probabilities and GLM model parameters are modelled jointly with prior restrictions put on the parameters that model the misclassification probabilities. The use of validation data has the advantage of avoiding distributional assumption of \( \gamma_0, \gamma_1 \), which would likely yield more flexible and realistic modelling of misclassification probabilities. However, validation data is rarely available and often costly to obtain. This is a current gap in methods that allow flexible modelling of misclassification probabilities without assuming the availability of validation data.
1.6 Empirical evaluation and comparison of methods

In this section, we include an extensive experimental comparison of existing methods described earlier on simulated data and real data that is part of the review paper to *Statistical Science* in preparation. The experimental investigation of the performance of different methods are obtained under various non-differential misclassification settings. We implement and compare six different methods: 1) EM of Magder and Hughes (1997), 2) post-fit adjustment of regression coefficients of Neuhaus (1999), 3) MC-SIMEX of Küchenhoff et al. (2006), 4) MEL of Rousseeuw and Christmann (2003), 5) two-step modelling of Lyles et al. (2011), and 6) MI + ML of Edwards et al. (2013). The first four methods assume known non-differential misclassification probabilities while the last two methods assume differential misclassification and the availability of validation data.

We first implement the methods on simulated datasets. We assume a logistic regression model with two covariates for the true responses, to which misclassification are introduced according to a non-differential contamination process. We then use the phoneme dataset - a dataset extracted from digitised speech from the TIMIT database and is used as a resource for speech recognition ([https://sci2s.ugr.es/keel/description.php](https://sci2s.ugr.es/keel/description.php)). There are 5405 observations, five covariates, and one binary response. The binary responses reflect with nasal vowel or oral vowel and 80% of the responses are nasal vowels.

We take the responses in phoneme data at hand as true responses and inject artificial contamination under a range of misclassification probabilities in the same way as the case with simulated data. The general setting used to create datasets with various levels of misclassification in the observed responses is described in Section 1.6.1 and the evaluation criteria employed to compare different methods is described in Section 4.2. Results from both simulated and phoneme data are presented and discussed in Section 1.6.4.
1.6.1 Set up

The set ups for the simulated and phoneme datasets in our experiments are similar. In simulated datasets, we generate true responses $t$ conditionally on covariate values of $X$ and in phoneme dataset, we take the responses at hand as true responses. For both datasets, artificial noise is then injected into $t$ by changing some of the 1’s and 0’s with probability $\gamma_1$ and $\gamma_0$ respectively, to create the observed responses $y$. A visualisation of the data generation process is shown in Figure 1.2.

Figure 1.2: Data generation process that illustrates the relationship between covariates $X$, true responses $t$, and observed responses $y$. In simulated datasets, we define a distribution on covariates $X$ and generate true responses $t$ conditionally on $X$ through a logistic regression model. In phoneme data, we take responses at hand as true responses. Individual positive and negative labels of $t$ are then ‘flipped’ with probability $\gamma_1$ and $\gamma_0$ respectively to generate observed responses $y$.

Simulated datasets

For the simulated datasets, we use two covariates for each observation, one continuous $x_{i1}$ and one binary $x_{i2}$. We simulate them by $x_{i1} \sim U(0,1)$ and $x_{i2} \sim Bern(0.5)$. To generate true responses $t$, we assume a logistic regression model for $t$ based on simulated $X$. Two different vectors of values are used as true regression coefficients for the logistic regression: $\beta^{\text{true}} = (-0.4, 0.4, 0.4)^T$ and $\beta^{\text{true}} = (-1.4, -2.2, -1)^T$. The first vector for $\beta^{\text{true}}$ leads to a proportion of 1/0 labels of the true responses at roughly 20/80 and the second vector at
about 50/50.

Six different levels of misclassification are applied to each set of true responses $t$, leading to six different sets of observed responses $y$. The set of $\gamma_0, \gamma_1$ pairs used are $(\gamma_0, \gamma_1) = (0.01, 0.01), (0.05, 0.05), (0.05, 0.1), (0.1, 0.1), (0.2, 0.1), (0.2, 0.2)$. We expect the strengths of each method to depend on factors like the level of misclassification in the observed responses and the balance of positive and negative labels in true responses.

In order to control for sampling variability, the covariates are fixed in each run of our experiments and the true responses $t$ and observed responses $y$ are generated using a set of pre-determined seeds. The same set of covariates is used in both vectors of $\beta^{true}$. For a given vector of true regression coefficients $\beta^{true}$, the experiment steps on simulated datasets
including the data generating process are summarised in Algorithm 7.

**Algorithm 7:** Experiments on simulated data

*Input:* Pre-determined coefficients $\beta^{\text{true}}$ that generated response by logistic regression; distribution for covariates $F$; number of total observations: $N$ and number of observations used for training: $n$;

*Output:* Estimates for $\beta, \beta_0$ and predictive statistics on test data;

1. Generate $N$ covariate vectors $x_1, ..., x_n, x_{n+1}, ..., x_N$ from $F$;
2. for $i \leftarrow 1$ to $R$ do
   3. Generate $n$ true responses $t_1, ..., t_n, t_{n+1}, ..., t_N$ according to the probabilities specified by the logistic regression and $\beta^{\text{true}}$;
   4. for $(\gamma_0, \gamma_1) \leftarrow (0.01, 0.01)$ to $(0.2, 0.2)$ do
      5. Flip some of labels among $t_1, ..., t_n$ according to $\gamma_0, \gamma_1$ to obtain observed responses $y_1, ..., y_n$;
      6. Apply the six methods on training set of each observed dataset composed of $(x_1, y_1), ..., (x_n, y_n)$ to obtain estimated regression coefficients;
      7. Calculate predictive statistics of various models trained on $(x_1, y_1), ..., (x_n, y_n)$ for test data;
   8. end
3. end

*Phoneme dataset with injected misclassification*

We use a real dataset where we adopt the responses at hand as true responses $t$. The same six different levels of misclassification are applied to $t$ to get six different sets of observed responses $y$. The data is the *phoneme* dataset extracted from digitised speech from the TIMIT database, a widely used resource for research in speech recognition (https://sci2s.ugr.es/keel/description.php). The dataset contains 5405 observations and five
covariates and all covariates are continuous attributes characterising each vowel. The covari- 
ates are Ene: aa, ao, dcl, iy, sh, which represent the amplitudes of the five first harmonics 
AHi, normalised by the total energy. The binary responses at hand correspond to the clas- 
sification of the class of the vowel with 0 for nasal vowel and 1 for oral vowels. In the data 
at hand, the majority of the observations (about 70%) are nasal vowels. Our choice of para-
metric model here is logistic regression regressing the responses on $X_1, X_2, X_3, X_4, X_5$. For 
comparison, we compare the estimated regression coefficients obtained from all six methods 
using training data with contaminated $y$ to the estimated regression coefficients obtained 
from the logistic regression regressing true response $t$ on $X_1, X_2, X_3, X_4, X_5$ using training 
data (which are used as the baseline for true values).

1.6.2 Implementation details

Some of the methods we have reviewed have accompanying available computer code, either 
for the entire method or for parts of the method. There is a link to SAS macro for using EM 
to iteratively estimate parameters of Magder and Hughes (1997) in the paper (though does 
not seem to be currently accessible). The MC-SIMEX method of Küchenhoff et al. (2006) 
can be implemented directly through R-package simex provided by Lederer et al. (2009). If 
all covariates are continuous and the data are not high dimensional, the robust distance 
for WEMEL can be computed using covMcd function in R package robustbase (Mächler, 
2019). On the other hand, for methods assume validation data, the SAS implementation 
of the method by Lyles et al. (2011) for using validation data to learn a model for true 
responses based on observed responses and covariates is provided in appendix of their paper. 
If one wants to use Edwards et al. (2013)’s approach of treating true responses in the main 
study data as missing, the mice function from R package mice by van Buuren et al. (2015) 
can be used to carry out multiple imputation. A summary of available softwares is listed in 
Table 1.2.
Table 1.2: Summary of available implementation tools for methods described in detail in this paper. Summary of methods described in detail in this paper where MEL is maximum estimated likelihood, ML is maximum likelihood.

In both simulated datasets and the phoneme dataset, data are split into training set and test set. In simulated datasets, there are 500 observations in the training sets and 100 observations in the test sets. In the phoneme dataset, 80% of the dataset is used as training set and the remaining 20% used test set. The results for accuracy of regression coefficient estimators are obtained from \((x_i, y_i)\) in the training sets and the results that reflect predictive power described in Section 4.2 are obtained from test sets. Our experiments for simulated datasets are repeated 1000 times with seeds generated beforehand for each run. For the phoneme dataset, since variation only exist for \(Y\) not \(T\), the experiments are repeated 100 times using 100 pre-generated seeds.

To implement the post-fit adjustment of regression coefficients, MEL and MC-SIMEX, non-differential misclassification is assumed and the values of \(\gamma_0\) and \(\gamma_1\) are required. In our implementation, the true values of \(\gamma_0\) and \(\gamma_1\) are assumed known and used. Calculating the magnitude of bias in Neuhaus (1999) also requires the true \(\beta_0\) and we use its true value in
our implementation. For EM and MEL, initial values of $\beta$ are needed and we set the initial guess for all parameters to be 0. Both the two-step modelling and MI + ML require some validation data where the true responses are known. We use a subset of the entire data as internal validation data and adopt the conventional setting of setting the size of the subset to 10% and 30% of all data.

All algorithms are implemented as outlined in the algorithms in Section 1.3 and Section 1.4. The `optim` package in R is used when some likelihood needs to be maximised, namely in post-fit adjustment of regression coefficients and MEL. The `brglm2` package (Kosmidis, 2017) implementing the bias-reducing adjusted scores method in Firth (1993) is used for fitting logistic regression to avoid infinite maximum likelihood estimates when the two classes are separable (more on conditions of separability are discussed in Appendix B.1 and a brief description of Firth’s corrected method is included in Appendix B.3). In other words, the places that we used `brglm2` to fit logistic regression are: the first step of Neuhaus (1999) that fits a logistic regression directly to observed data, the first step of Lyles et al. (2011) to learn $p(y_i = 1 \mid t_i, x_i)$ from the validation data, and the first step of Edwards et al. (2013) to learn $p(t_i = 1 \mid y_i, x_i)$ from the validation data. The MC-SIMEX method of Küchenhoff et al. (2006) is implemented directly through the R package `simex` provided by Lederer et al. (2009). In the simulation step of MC-SIMEX, $(0.5, 1, 1.5, 2)$ are used for $\lambda$ and 100 pseudo datasets are generated for each $\lambda$. In the extrapolation step, quadratic regression is chosen as the method of extrapolation. Technically, the steps of fitting directly observed data and simulated data might also have separability issues, but the `simex` package only operates on `glm` class object to fit a regular logistic regression. In the EM algorithm, we consider the algorithm converged if the difference between two rounds of estimates of the parameters is less than 0.01. The algorithm runs until convergence or reaching a maximum iteration of 1000. For MI+ML, the multiple imputation part is carried out by `mice` function from R package `mice` by van Buuren et al. (2015) and we used 20 imputed datasets in our
implementation.

1.6.3 Evaluation criteria

We are interested in investigating how well each method estimates the true regression coefficients and predicts out-of-sample observations. We assess inference results by the distribution of estimated coefficients obtained for the logistic regression trained on training data through various methods. To evaluate predictive power, we use the typical criterion of classification accuracy (or balanced accuracy), in which the proportion of correctly classified observations is calculated among all observations. In addition to balanced accuracy, we also want to evaluate the accuracy within each class, therefore also adopt the AUC and F-measure.

The *Area under the ROC Curve* (AUC) provides a single numeric measure bounded between 0 and 1 that summarises the tradeoff between sample true positive rate (TPR) and sample false positive rate (FPR) over all thresholds (James et al., 2013, chap. 4). Sample TPR describes how good the model is at predicting the positive class when the actual output is positive and the sample FPR summarises how often a positive class is predicted when the actual outcome is negative. Mathematically, this can be written as

\[
TPR = \frac{TP \text{ (true positives)}}{TP + FN \text{ (false negatives)}} = \text{sensitivity = recall},
\]

\[
FPR = \frac{FN}{FP \text{ (false positives)} + TN \text{ (true negatives)}} = 1 - \text{specificity}.
\]

The *Receiver Operating Characteristics* (ROC) curve visually summarises the trade off between sample TPR (y-axis) and sample FPR (x-axis) at different thresholds (James et al., 2013, chap. 4) with the optimal \((TPR, FPR)\) at \((1, 0)\), corresponding to the left corner of the ROC curve. The more the ROC curve of a model approaches the left corner, the better the method is according to the ROC curve. Because we repeat our experiment many times,
it is hard to visualise ROC curve as we have one curve for each setting in each run. The AUC, on the other hand, is a single numeric measure and we can compute its mean and standard deviation from many iterations.

Note that the sample TPR is also known as sensitivity or recall. Recall is frequently used along with precision which describes how good the model is at predicting the positive class in our sample and can be calculated by

\[
\text{Precision} = \frac{TP}{TP + FP}.
\]

The F-measure is a composite measure combining precision and recall and penalising extreme values (Chinchor, 1992). In the most commonly-used form of F-measure (in which it is the harmonic mean of precision and recall), recall is weighted twice as important as precision, expressed by

\[
\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]

While the ROC curve is good for evaluating classification performance in balanced datasets, F-measure is a better evaluation criterion for imbalanced data where the majority of the observations are negatives. This can be contributed to the use of TN in FPR (thus in ROC curve and AUC) but not in F-measure. Large number of TN would drive FPR to be smaller, which consequently lead to high AUC.

1.6.4 Results

We present selective results of experimental studies on the simulated datasets of 1000 repeated runs and the phoneme dataset of 100 repeated runs. These include the estimated regression coefficients of the logistic regression obtained from training data using all six methods (along with fitting logistic regression directly to the true responses \( t \) and to the
observed responses $y$ for comparison) and results reflecting predictive power obtained from test data. In the simulated datasets, the test data are the 100 observations whose responses are not contaminated. In the phoneme dataset, 20% of the data are first selected as test set with uncontaminated responses. Though the simulated datasets from two different vectors of $\beta^{true}$ and the phoneme dataset have different covariates structure and different proportions of 0/1 labels, the results of various models’ inferential and predictive power are similar.

As shown in Figure 1.4, Table 1.4, Figure 1.6, and Figure 1.8, for both the simulated and phoneme datasets, in the low misclassification case where $(\gamma_0, \gamma_1) = (0.01, 0.01)$, all methods obtain estimates of regression coefficients close to the true values. As misclassification probabilities increase, EM, two-step modelling, and MI+ML generally still do well in recovering the true GLM regression coefficients. However, as suggested by Table 1.3, the two-step modelling and MI+ML might not always produce good results especially when validation data is only 10% of all data. In both methods, there are bias in the mean values of estimated regression coefficients in some misclassification settings. This is not surprising as we expect the fit of these two models to depend on the quality of the validation data used in the first step of fitting a logistic regression to learn $p(y_i = 1 \mid t_i, x_i)$ or $p(t_i = 1 \mid y_i, x_i)$. In some cases, the validation data has separation issue and we use the brglm2 package of Kosmidis (2017) to learn the logistic regression model for $p(y_i = 1 \mid t_i, x_i)$ in the two-step modelling. The use of the bias-reducing adjusted scores method avoids infinite estimates but sometimes gives large values of estimated $\beta$’s. Therefore we see large (in magnitude) estimated mean values and standard deviation in Table 1.3, even though the median estimated values for $\beta$’s in these situations has are close to true values as shown in Figure 1.6. The post-fit adjustment of regression coefficients also gives good results in the balanced datasets where the true regression coefficients are close to 0 (which is an underlying assumption of the Taylor expansion explained in Section 1.3.2 that is used to obtain the closed-form expression of the magnitude of bias). MEL is found to provide biased estimates of regression coefficients when
the misclassification probabilities is not low (above 0.01) and so does MC-SIMEX when the misclassification probabilities are moderately high (at least 0.1).

Figure 1.3: Comparison of the estimates (denotes average) of $\beta_0$ in simulated dataset in exp 1 where the proportion of 1/0 are roughly balanced with $(\beta_0, \beta_1, \beta_2)^{\text{true}} = (-0.4, 0.4, 0.4)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.
Figure 1.4: Comparison of the estimates (denotes average) of $\beta_1$ in simulated dataset in exp 1 where the proportion of 1/0 are roughly balanced with $(\beta_0, \beta_1, \beta_2)^{true} = (-0.4, 0.4, 0.4)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.
<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{\beta}_0 ) (sd)</th>
<th>( \hat{\beta}_1 ) (sd)</th>
<th>( \hat{\beta}_2 ) (sd)</th>
<th>( \hat{\beta}_0 ) (sd)</th>
<th>( \hat{\beta}_1 ) (sd)</th>
<th>( \hat{\beta}_2 ) (sd)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>-0.395 (0.132)</td>
<td>0.408 (0.092)</td>
<td>0.389 (0.185)</td>
<td>-0.395 (0.129)</td>
<td>0.408 (0.092)</td>
<td>0.389 (0.185)</td>
</tr>
<tr>
<td>EM</td>
<td>-0.398 (0.131)</td>
<td>0.411 (0.096)</td>
<td>0.391 (0.190)</td>
<td>-0.395 (0.144)</td>
<td>0.411 (0.105)</td>
<td>0.384 (0.211)</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>-0.400 (-)</td>
<td>0.407 (0.094)</td>
<td>0.388 (0.189)</td>
<td>-0.406 (-)</td>
<td>0.405 (0.102)</td>
<td>0.388 (0.209)</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>-0.398 (0.132)</td>
<td>0.411 (0.096)</td>
<td>0.391 (0.191)</td>
<td>-0.394 (0.146)</td>
<td>0.410 (0.106)</td>
<td>0.384 (0.213)</td>
</tr>
<tr>
<td>MEL</td>
<td>-0.380 (0.125)</td>
<td>0.393 (0.091)</td>
<td>0.374 (0.182)</td>
<td>-0.313 (0.114)</td>
<td>0.325 (0.081)</td>
<td>0.304 (0.167)</td>
</tr>
<tr>
<td>Two-step modelling (10% V)</td>
<td>-0.388 (0.141)</td>
<td>0.407 (0.271)</td>
<td>0.395 (0.499)</td>
<td>-0.425 (0.334)</td>
<td>0.464 (0.277)</td>
<td>0.415 (0.477)</td>
</tr>
<tr>
<td>Two-step modelling (30% V)</td>
<td>-0.393 (0.162)</td>
<td>0.413 (0.129)</td>
<td>0.394 (0.236)</td>
<td>-0.405 (0.222)</td>
<td>0.436 (0.176)</td>
<td>0.403 (0.339)</td>
</tr>
<tr>
<td>MI + ML (10% V)</td>
<td>-0.414 (0.208)</td>
<td>0.452 (0.176)</td>
<td>0.415 (0.290)</td>
<td>-0.383 (0.246)</td>
<td>0.423 (0.212)</td>
<td>0.398 (0.365)</td>
</tr>
<tr>
<td>MI + ML (30% V)</td>
<td>-0.405 (0.176)</td>
<td>0.412 (0.142)</td>
<td>0.404 (0.263)</td>
<td>-0.392 (0.162)</td>
<td>0.415 (0.118)</td>
<td>0.392 (0.233)</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>-0.386 (0.128)</td>
<td>0.398 (0.092)</td>
<td>0.386 (0.185)</td>
<td>-0.349 (0.127)</td>
<td>0.361 (0.091)</td>
<td>0.340 (0.186)</td>
</tr>
</tbody>
</table>

\((a) (\gamma_0, \gamma_1) = (0.01, 0.01)\)

\((b) (\gamma_0, \gamma_1) = (0.05, 0.05)\)

\((c) (\gamma_0, \gamma_1) = (0.05, 0.1)\)

\((d) (\gamma_0, \gamma_1) = (0.1, 1)\)

\((e) (\gamma_0, \gamma_1) = (0.2, 0.1)\)

\((f) (\gamma_0, \gamma_1) = (0.2, 0.2)\)

Table 1.3: Comparison of the average estimates of \(\beta_0, \beta_1, \beta_2\) under different \((\gamma_0, \gamma_1)\) when the proportion of 1/0 are roughly balanced with \((\gamma_0, \beta_1, \beta_2)^{true} = (-0.4, 0.4, 0.4)^T\) and experiments are repeated 1000 times under pre-determined seeds.
Figure 1.5: Comparison of the estimates ($\hat{\gamma}$ denotes average) of $\beta_1$ in simulated dataset in exp 2 where the proportion of 1/0 are roughly 20/80 with $(\beta_0, \beta_1, \beta_2) = (-1.4, -2.2, -1)^T$ and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The dash lines represent the true values of the parameters.
Figure 1.6: Comparison of the estimates (\( \hat{\gamma} \) denotes average) of \( \beta_2 \) in simulated dataset in exp 2 where the proportion of 1/0 are roughly 20/80 with \((\beta_{0}^{true}, \beta_{1}^{true}, \beta_{2}^{true}) = (-1.4, -2.2, -1)^T\) and misclassified responses generated from true ones under different \((\gamma_0, \gamma_1)\). The dash lines represent the true values of the parameters.
Table 1.4: Comparison of the average estimates of $\beta_0, \beta_1, \beta_2$ in exp 2 under different $(\gamma_0, \gamma_1)$ when the proportion of 1/0 are roughly 20/80 with $(\beta_0, \beta_1, \beta_2)^{\text{true}} = (-1.4, -2.2, -1)^T$ and experiments are repeated 1000 times under pre-determined seeds. Note that even though the median estimated values for $\beta$'s using the two-step modelling seem to have extreme mean estimates, the median estimates are actually close to true values as shown in Figure 1.6. The large mean estimates are due to separation issue in validation data and we use the brglm2 package of Kosmidis (2017) to learn the logistic regression model for $p(y_i = 1 \mid t_i, x_i)$ in validation data. The use of the bias-reducing adjusted scores method avoids infinite estimates but sometimes give large values of estimated $\beta$'s.
(a) \((\gamma_0, \gamma_1) = (0.01, 0.01)\)

(b) \((\gamma_0, \gamma_1) = (0.05, 0.05)\)

(c) \((\gamma_0, \gamma_1) = (0.05, 0.1)\)

(d) \((\gamma_0, \gamma_1) = (0.1, 0.1)\)

(e) \((\gamma_0, \gamma_1) = (0.2, 0.1)\)

(f) \((\gamma_0, \gamma_1) = (0.2, 0.2)\)

Figure 1.7: Comparison of the estimates (\(\hat{\gamma}\) denotes average) of \(\beta_0\) in phoneme dataset experiment where the proportion of 1/0 are roughly 30/70 with \((\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (-1.065, -0.610, -0.406, 0.672, 0.788, 0.541)^T\) and misclassified responses generated from true ones under different \((\gamma_0, \gamma_1)\). The dash lines represent the true values of the parameters.
Figure 1.8: Comparison of the estimates (\( \checkmark \) denotes average) of \( \beta_4 \) in phoneme dataset where the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true ones under different \( (\gamma_0, \gamma_1) \). The dash lines represent the true values of the parameters.

For simulated datasets and simulated phoneme dataset, using EM to iteratively estimate parameters of logistic regression (Magder and Hughes, 1997), building an additional logistic regression on validation data to learn misclassification probabilities (Lyles et al., 2011), and treating \( t \) in main study group as missing and create multiple imputed datasets by modelling observations in the validation group (Edwards et al., 2013) generally do well in recovering the
Table 1.5: Comparison of the average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$ under different $(\gamma_0, \gamma_1)$ for the phoneme dataset where the proportion of I/O are roughly 70/30, $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (-1.065, -0.610, -0.406, 0.672, 0.788, 0.541)^T$ and experiments are repeated 1000 times under pre-determined seeds.

(a) $(\gamma_0, \gamma_1) = (0.01, 0.01)$

(b) $(\gamma_0, \gamma_1) = (0.05, 0.05)$

(c) $(\gamma_0, \gamma_1) = (0.05, 0.1)$

(d) $(\gamma_0, \gamma_1) = (0.1, 0.1)$

(e) $(\gamma_0, \gamma_1) = (0.2, 0.1)$

(f) $(\gamma_0, \gamma_1) = (0.2, 0.2)$
true GLM regression coefficients. However, as suggested by Table 1.3 and Table 1.4, building an additional logistic regression on validation data to learn misclassification probabilities (Lyles et al., 2011), and treating t in main study group as missing and create multiple imputed datasets by modelling observations in the validation group (Edwards et al., 2013) might not always produce stable good results especially when there is only 10% of the data as validation data, as shown by bias in the mean values of regression coefficients in some misclassification settings. This is not surprising as we expect the fit to depend on the quality of the validation data used in the first step of fitting a LR to learn \( p(y_i = 1 \mid t_i, x_i) \) and \( p(t_i = 1 \mid y_i, x_i) \). Adjusting regression coefficients by the magnitude of bias (Neuhaus, 1999) also gives good results in the balanced dataset where the true regression coefficients are close to 0. In fact, in the low misclassification case where \((\gamma_0, \gamma_1) = (0.01, 0.01)\), all methods obtain estimates of regression coefficients close to the true values. However, as the amount of misclassification increases, maximising estimated likelihood using pseudo-observations (Rousseeuw and Christmann, 2003) and simulating more data with added error to learn error trend (Küchenhoff et al., 2006) starts to give biased estimates of regression coefficients.

Looking at results reflecting predictive power for both simulated datasets and the phoneme dataset, the differences among various methods is very small compared to differences in estimated coefficients. This could be due to the fact that we get the same ranking of the six methods by using AUC, accuracy, and F-measure as evaluation criteria compared to using estimated regression coefficients. We report the results for accuracy and F-measure for exp 2 and phoneme experiment in Figure 1.11 and Figure 1.13. The AUC results display similar characteristics where the values from different methods are close with similar standard deviation (figures can be found in Supplementary material). This is not surprising as examining individual ROC curves across different methods in a particular experiment run, they are largely similar. Similar results has also been reported in Rantalainen and Holmes (2011).
Figure 1.9: Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in exp 1 using various methods where the data are simulated and the proportion of 1/0 are roughly 50/50 and misclassified responses generated from true responses under different $(\gamma_0, \gamma_1)$. The boxplots represent the accuracy and F-measure from 1000 runs and bars represent their standard deviations.
Figure 1.10: Comparison of AUC evaluated on test data in exp 1 using various methods where the data are simulated and the proportion of 1/0 are 50/50 and misclassified responses are generated from true responses under different \((\gamma_0, \gamma_1)\). The boxplots represent the AUC from 1000 runs and bars represent their standard deviations.
Figure 1.11: Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in exp 2 using various methods where the data are simulated and the proportion of 1/0 are 20/80 and misclassified responses are generated from true responses under different \((\gamma_0, \gamma_1)\). The boxplots represent the accuracy and F-measure from 1000 runs and bars represent their standard deviations.
Figure 1.12: Comparison of AUC evaluated on test data in exp 2 using various methods where the data are simulated and the proportion of 1/0 are 20/80 and misclassified responses are generated from true responses under different $(\gamma_0, \gamma_1)$. The boxplots represent the AUC from 1000 runs and bars represent their standard deviations.
Figure 1.13: Comparison of accuracy and F-measure evaluated on test data at 0.5 threshold in phoneme dataset experiment using various methods where the data are simulated and the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true responses under different $(\gamma_0, \gamma_1)$. The boxplots represent the accuracy and F-measure from 100 runs and bars represent their standard deviations.
Figure 1.14: Comparison of AUC evaluated on test data using various methods in phoneme dataset experiment where the proportion of 1/0 are roughly 30/70 and misclassified responses generated from true ones under different $(\gamma_0, \gamma_1)$. The boxplots represent the AUC from 100 runs and bars represent their standard deviations.

### 1.6.5 Advantages and disadvantages

Note that the first step of Lyles et al. (2011) builds a logistic regression model regressing $y$ on $t$ and $X$, which naturally implies the underlying assumption that misclassification are differential. If we want to explicitly implement non-differential misclassification, we should
not use any covariates in equation (1.23) and use \( \eta_{it} = \logit[p(y_i = 1 \mid t_i = t)] = \alpha_0 + \alpha_1 t \) instead. This also applies to the likelihood \( L_S \) and \( L_V \). Generally speaking, everywhere we currently have \( \Gamma_0(x_i) \) and \( \Gamma_1(x_i) \), we should have \( \gamma_0 \) and \( \gamma_1 \) instead to reflect that we are explicitly assuming non-differential misclassification.

In our experiments, we implement what we have described in Algorithm 5, which assumes differential misclassification. The model should still be able to learn non-differential misclassification probabilities automatically by getting estimates of \( \alpha_2, \ldots, \alpha_p \) all close to 0 since non-differential misclassification is a special case of differential misclassification. This practice is likely to be prone to overfitting. Like in Lyles et al. (2011), the use of \( \gamma_0, \gamma_1 \) in the approach of Katz and Katz (2010) naturally implies differential misclassification. Validation data is mainly used to allow borrowing of coefficients that models misclassification probabilities and could be data obtained from auxiliary sources. For Edwards et al. (2013), the initial motivation was a case-control study and the main interest was the odds ratio estimates. Nevertheless, the approach of Edwards et al. (2013) can be applied when the inference interest is regression coefficients of a pre-specified model.
Chapter 2

Bayesian Nonparametrics

This chapter gives an overview of Bayesian nonparametrics, especially those concepts that are useful for Bayesian nonparametric mixture models. Bayesian nonparametric models are models over infinite dimensional spaces, e.g. of functions, densities. They are used in a variety of problems, regression, classification, clustering, latent variable modeling, sequential modeling (Orbanz and Teh, 2010). The parameter space of Bayesian nonparametric models is typically chosen as the set of all possible solutions for a given learning problem. For example, in a regression problem the parameter space can be the set of continuous functions, and in a density estimation problem the space can consist of all densities (Orbanz and Teh, 2010).

A Bayesian nonparametric model uses only a finite subset of the available parameter dimensions to explain a finite sample of observations with the set of dimensions chosen depending on the sample. One major advantage is that the effective complexity of the model adapts to the data (Orbanz and Teh, 2010). Therefore, classical adaptive problems such as nonparametric estimation and model selection can be formulated as Bayesian inference problems. We focus our discussions on Dirichlet process mixture models, which provide natural setting for density estimation. The number of mixture components in Dirichlet process mixture models adapts to the complexity of the data and simultaneously estimate
both the number of components in mixture model and the parameters of individual mixture components (Escobar and West, 1995).

Before we can describe Dirichlet process mixture models, we layout some notations and useful definitions that will be used later in Section 2.1, Section 2.2, and Section 2.3. In Section 2.4, we define Dirichlet distribution and discuss its properties and common usage. Section 2.5 defines Dirichlet process and the stick-breaking construction, which is used in proposed model in Chapter 3. Section 2.6 discusses some applications of Dirichlet process and introduces Dirichlet process mixture models. Section 2.7, Section 2.8 and Section 2.9 review MCMC basics and discuss the posterior sampling of Dirichlet process mixture models.

2.1 Notations and useful definitions

In this section, we review some terminologies and mathematical definitions that are later used to define Dirichlet process.

A \( \sigma \)-algebra \( \mathcal{A} \) on a set \( X \) is a family of subsets \( X \) with the following properties

\[
\begin{align*}
X & \in \mathcal{A} \\
A & \in \mathcal{A} \Rightarrow A^c \in \mathcal{A} \\
(A_j)_{j \in \mathbb{N}} & \in \mathcal{A} \Rightarrow \bigcup_{j \in \mathbb{N}} (A_j) \in \mathcal{A}.
\end{align*}
\]

A set \( A \in \mathcal{A} \) is said to be \( \mathcal{A} \)-measurable (Schilling, 2005).

The \( \sigma \)-algebra \( \sigma(\mathcal{O}^n) \) generated by the open sets \( \sigma(\mathcal{O}^n) \) of \( \mathbb{R}^n \) is called Borel \( \sigma \)-algebra and its members are called Borel (measurable) sets. We use \( \mathcal{B}(\mathbb{R}^n) \) or \( \mathcal{B}^n \) to denote Borel sets in \( \mathbb{R}^n \) (Schilling, 2005).

A (positive) measure \( \mu \) on \( X \) is a mapping \( \mu : \mathcal{A} \to [0, \infty] \) defined on a \( \sigma \)-algebra \( \mathcal{A} \)
satisfying $\mu(\sigma) = 0$ and for any countable family of pairwise disjoint sets $(A_j)_{j \in \mathbb{N}} \in \mathcal{A}$

$$\mu\left(\bigcup_{j \in \mathbb{N}}(A_j)\right) = \sum_{j \in \mathbb{N}} \mu(A_j).$$

Let $X$ be a set and $\mathcal{A}$ be $\sigma$-algebra on $X$. The pair $(X, \mathcal{A})$ is called measurable space (Schilling, 2005). If $\mu$ is a measure on $X$, then $(X, \mathcal{A}, \mu)$ is called measure space (Schilling, 2005). A finite measure is a measure with $\mu(X) < \infty$ and a probability measure is a measure with $\mu(X) = 1$ (Schilling, 2005).

Let $(X, \mathcal{A})$ be any measurable space and let $x \in X$ be some point. The Dirac (delta) measure $\delta_x$ is defined for $A \in \mathcal{A}$ by

$$\delta_x(A) = \begin{cases} 
1 & \text{if } x \in A, \\
0 & \text{if } x \notin A.
\end{cases}$$

Dirac (delta) measure $\delta_x$ is also referred to as the unit mass (point mass) at point $x$ (Schilling, 2005).

A set $A$ in $\sigma$-algebra $\mathcal{A}$ is called an atom if there is no proper subset $B \subsetneq A$ such that $B \in A$. A measure is called atomic if every measurable set of positive measure contains an atom (Schilling, 2005). Note that a bounded positive measure $\mu$ on a measurable space $(X, \Sigma)$ is atomic if and only if it is a weighted sum of countably many Dirac measures (Aldous, 1985).

Next, we briefly discuss Gamma and Beta distribution which are related to Dirichlet distribution. Then, we introduce Dirichlet distribution, of which Dirichlet process can be thought as an infinite-dimensional generalisation. We describe some properties of Dirichlet distribution and Dirichlet process that may seem irrelevant at glance but are crucial to the
sampling algorithm of Dirichlet process mixture models.

2.2 Gamma distribution

The probability density function *Gamma distribution* for \( x \in [0, \infty] \) where \( x \sim \text{Gamma}(\alpha, \beta) \) is defined as

\[
f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}
\]

where \( \alpha > 0 \) is often referred to as the shape parameter and \( \beta > 0 \) the inverse scale parameter (Gelman et al., 2013). It has expectation and variance following

\[
\mathbb{E}(x) = \frac{\alpha - 1}{\beta}, \quad \text{Var}(x) = \frac{\alpha}{\beta^2}.
\]

2.3 Beta distribution

The probability density function for *Beta distribution* for \( 0 \leq x \leq 1 \) where \( x \sim \text{Beta}(\alpha, \beta) \) where shape parameter \( \alpha > 0, \beta > 0 \) follows

\[
f(x|\alpha, \beta) = \frac{1}{\text{Beta}(\alpha, \beta)} x^{\alpha-1} (1 - x)^{\beta-1}
\]

where \( \text{Beta}(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \) (Gelman et al., 2013). It has expectation and variance following

\[
\mathbb{E}(x) = \frac{\alpha}{\alpha + \beta}, \quad \text{Var}(x) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.
\]

The expectation of the logarithm of Beta distributed \( x \) follows

\[
\mathbb{E}[\log(x)] = \psi(\alpha) - \psi(\alpha + \beta)
\]

\[
\mathbb{E}[\log(1 - x)] = \psi(\beta) - \psi(\alpha + \beta)
\]

(2.1)
where \( \psi(\cdot) \) denote the digamma function (Robert and Casella, 2013) and its derivation is shown in Appendix C.1. The digamma function is defined as the logarithmic derivative of the Gamma function where

\[
\psi(x) = \frac{d}{dx} \log(\Gamma(x)) = \frac{\Gamma'(x)}{\Gamma(x)}.
\]

In Bayesian inference, the Beta distribution is a conjugate prior for Bernoulli, binomial, negative binomial, and geometric distribution. Thus, the Beta distribution is a convenient prior for percentages and proportions where the hyperparameters of Beta can be set by fixing its mean and variance (Gelman et al., 2013).

### 2.4 Dirichlet distribution

The *Dirichlet distribution* is the \((k-1)\)-variate analogue of Beta distribution with the probability density function

\[
f(x_1, \ldots, x_{k-1}; \alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_k)} x_1^{\alpha_1-1} \cdots x_{k-1}^{\alpha_{k-1}}
\]

at any point in the simplex where \( \alpha_i \) are all positive, \( x_i \geq 0 \) for \( i = 1, \ldots, k \) and \( \sum_{i=1}^{k} x_i = 1 \). A distribution having the above probability density function is known as the \((k-1)\)-variate Dirichlet distribution \( D(\alpha_1 + \ldots + \alpha_{k-1}; \alpha_k) \) (Wilks, 1962). Note that the Beta distribution is the special case of Dirichlet distribution when \( k = 2 \), which is equivalent to \( x_1 \sim \text{Beta}(\alpha_1, \alpha_2) \) and \( x_2 = 1-x_1 \). The sum of the random variables \( x_1 + \ldots + x_{k-1} \) also follows Beta distribution \( \text{Beta}(\alpha_1 + \ldots + \alpha_{k-1}, \alpha_k) \).

A general form for moment of the \( k \)-variate Dirichlet distribution is in Wilks (1962) and
its means, variance, and covariances can be written as

\[ \mu(x_i) = \frac{\alpha_i}{\alpha_1 + \ldots + \alpha_k}, \]

\[ \text{Var}(x_i) = \frac{\alpha_i(\alpha_1 + \ldots + \alpha_k - \alpha_i)}{(\alpha_1 + \ldots + \alpha_k + 1)^2}, \]

\[ \text{Cov}(x_i, x_j) = -\frac{\alpha_i\alpha_j}{(\alpha_1 + \ldots + \alpha_k + 1)^2} (i \neq j). \]

Note that the higher the \( \alpha_i \) are, the smaller the variance. If we multiply \( \alpha \) by a factor less than 1, we would result in the same mean but smaller variance and covariance.

### 2.4.1 Conjugacy with the multinomial distribution

A key property of the Dirichlet distribution that makes it especially useful in the Bayesian context is its **conjugacy with the multinomial distribution**. Suppose we take \( n \) independent samples from \( k \) mutually exclusive categories where the \( i \)th category has success probability \( p_i \). The number of occurrences for category \( i \), \( w_i \), follows multinomial distribution \( \text{multinomial}(n; p_1, \ldots, p_k) \) has probability mass function

\[ f(w_1, \ldots, w_k|n; p_1, \ldots, p_k) = \frac{n!}{w_1!w_2!\ldots w_k!} \prod_{i=1}^{k} p_i^{w_i}. \]

Dirichlet distribution is the conjugate distribution for multinomial distribution in the sense that if the probability of category vector, \( \mathbf{p} = (p_1, \ldots, p_k) \), follows \( \mathbf{p} \sim \text{Dir}(\alpha_1, \ldots, \alpha_k) \) with \( \mathbf{x}|\mathbf{p} \sim \text{multinomial}(n; p_1, \ldots, p_k) \), then the posterior distribution is given by \( \mathbf{p}|\mathbf{x} \sim \text{Dir}(\alpha_1 + w_1, \ldots, \alpha_k + w_k) \).

The Dirichlet distribution is not only conjugate with the multinomial distribution, its marginal distribution is also a Dirichlet distribution. Suppose a random vector \( (x_1, \ldots, x_k) \) follows \( k \)-variate Dirichlet distribution \( \text{Dirichlet}(\alpha_1, \ldots, \alpha_{k-1}, \alpha_k) \), then the marginal distribu-
tion of \((x_1, \ldots, x_l)\) where \(l < k\) follows \(l\)-variate Dirichlet distribution \(\text{Dirichlet}(\alpha_1, \ldots, \alpha_l, \alpha_{l+1} + \ldots + \alpha_k)\) (Wilks, 1962).

These two properties mentioned above make the Dirichlet distribution an attractive prior in Bayesian models. In Bayesian mixture models of \(k\) components, \((k - 1)\)-variate Dirichlet distribution is a natural choice for the prior for mixing proportions \(p_1, \ldots, p_{k-1}\) because the constraint that all mixing proportions should add up to 1 is automatically satisfied by the definition of Dirichlet distribution which specifies a probability density function for any point in the simplex.

### 2.4.2 Other properties

The Dirichlet distribution also possesses some other nice properties that can be seen via further examination of the relationship between Dirichlet and Gamma distribution. These properties show that the manipulation of Dirichlet distribution often lead to another Dirichlet distribution, making Dirichlet distribution a good choice for priors.

**Normalisation rule**: Suppose \(\eta_1, \ldots, \eta_k\) are independent random variables having Gamma distribution \(\text{Gamma}(\alpha_1, \lambda), \ldots, \text{Gamma}(\alpha_k, \lambda)\), then their normalised sum also follows a Dirichlet distribution (Wilks, 1962). Mathematically, we can write

\[
\frac{(\eta_1, \ldots, \eta_k)}{\eta_1 + \ldots + \eta_k} \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k).
\]

In other words, the normalisation rule tells us that we can construct Dirichlet distribution by normalising independent random variables following Gamma distribution with the same scale parameter. By the property of Gamma distribution, we have \(\eta_1 \sim \text{Gamma}(\alpha_1, \lambda), \eta_2 \sim \text{Gamma}(\alpha_2, \lambda)\) implies \(\eta_1 + \eta_2 \sim \Gamma(\alpha_1 + \alpha_2, \lambda)\). Therefore, if \((\pi_1, \ldots, \pi_k) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k),\)
assuming \( \pi_i \) is the normalised \( \eta_i \) that follows \( \Gamma(\alpha_i, \lambda) \) for \( i = 1, ..., k \), we can get

\[
(\pi_1 + \pi_2, \pi_3, ..., \pi_k) = \left( \frac{\eta_1 + \eta_2}{\sum_{i=1}^k \eta_i}, \frac{\eta_3}{\sum_{i=1}^k \eta_i}, ..., \frac{\eta_k}{\sum_{i=1}^k \eta_i} \right) \sim \text{Dirichlet}(\alpha_1 + \alpha_2, \alpha_3, ..., \alpha_k).
\]

by the normalisation rule. This property is known as the **agglomeration rule** (Zhang, 2008). A general version states that if \((I_1, ..., I_s)\) is a partition of \(\{1, ..., k\}\) where \((\pi_1, ..., \pi_k) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_k)\), then

\[
\left( \sum_{i \in I_1} \pi_i, ..., \sum_{i \in I_s} \pi_i \right) \sim \text{Dirichlet} \left( \sum_{i \in I_1} \alpha_i, ..., \sum_{i \in I_s} \alpha_i \right)
\]

The agglomeration rule enables the reduction of the dimensionality of a Dirichlet distribution.

Another helpful property of Dirichlet distribution is the **expansion rule**, which allows the dimensionality of a Dirichlet distribution to be increased. Suppose \((\pi_1, ..., \pi_k) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_k)\), if \((\theta_1, ..., \theta_m) \sim \text{Dirichlet}(\alpha_1 b_1, \alpha_1 b_2, ..., \alpha_1 b_m)\) with \(\sum_{i=1}^m b_i = 1\), then

\[
(\pi_1 \theta_1, ..., \pi_1 \theta_m, \pi_2, ..., \pi_k) \sim \text{Dirichlet}(\alpha_1 b_1, \alpha_1 b_2, ..., \alpha_1 b_m, \alpha_2, ..., \alpha_k).
\]

The proof of the expansion rule replies on construction of \(z_1 \sim \Gamma(\alpha_1 b_1, 1), ..., z_m \sim \Gamma(\alpha_1 b_m, 1)\), analogous to the case of the agglomeration rule.

### 2.5 Dirichlet process

The **Dirichlet process** can be thought as an infinite-dimensional generalisation of Dirichlet distribution that can be used as a prior on unknown distributions (Gelman et al., 2013). Dirichlet process provides a natural way to extend finite component mixture models to infinite component mixture models (Gelman et al., 2013).
Suppose $\Omega$ is sample space, $\mathcal{B}$ is the collection of all possible subsets $\Omega$, and $P_0$ is a probability measure on the space. Let $\alpha$ be a positive real number, a Dirichlet process $DP(\alpha, P_0)$ is defined to be the distribution of a random probability measure $P$ over $(\Omega, \mathcal{B})$ such that for any finite measurable subsets $B_1, ..., B_r$ random vector $(P(B_1), ..., P(B_r))$ is distributed as a finite-dimension Dirichlet distribution where

$$P(B_1), ..., P(B_r) \sim Dirichlet(\alpha P_0(B_1), ..., \alpha P_0(B_r)).$$

$P_0$ is referred to as the base measure of the Dirichlet process and $\alpha$ the prior concentration parameter (Teh et al., 2005). The definition of Dirichlet Process is first given in (Ferguson, 1973), who also proves the existence of such random probability measure $P$ since we need $P$ to exist to assign probabilities to any measurable partitions $B_1, ..., B_r$. $P \sim DP(\alpha P_0)$ is often used as concise notation to indicate that a probability measure $P$ on $(\Omega, \mathcal{B})$ is assigned a Dirichlet process (DP) prior. A Dirichlet process $DP(\alpha P_0)$ on $(\Omega, \mathcal{B})$ has its mean and variance given by

$$\mathbb{E}(P(B)) = P_0(B), \quad Var(P(B)) = \frac{P_0(B)(1 - P_0(B))}{\alpha + 1}$$

where $B \in \mathcal{B}$ (Gelman et al., 2013). In $P \sim DP(\alpha P_0)$, $P_0$ acts like a mean parameter where the prior for $P$ is centered on and $\alpha$ can be viewed as a precision parameter, controlling the degree of shrinkage of $P$ towards $P_0$ (Teh, 2010).

### 2.5.1 Conjugacy property

One thing that make DP a nice prior for Bayesian nonparametrics is that DP provides a conjugate family of priors over distributions that is closed under posterior updates given observations (Teh, 2010). In short, the conjugacy property of DP says that if $\theta = (\theta_1, ..., \theta_n)$
is an independent and identically distributed (i.i.d) sample with $\theta_i | P \sim P$ and $P \sim DP(\alpha P_0)$, then given observed values $\theta$

$$P(\theta) \sim DP(\alpha + n, P_1)$$

where $P_1 \propto \alpha P_0 + \sum_{i=1}^n \delta_{\theta_i}$ (Teh, 2010).

The conjugacy property of Dirichlet process can be seen from the conjugacy of Dirichlet and multinomial distributions discussed in Section 2.4.1. Let $\theta_i \sim P$ for $i = 1, \ldots, n$ and $P \sim DP(\alpha P_0)$, by the conjugacy of Dirichlet and multinomial distributions, for any measurable finite partition $B_1, \ldots, B_r$, we have

$$P(B_1), \ldots, P(B_r)|\theta_1, \ldots, \theta_n \sim Dirichlet\left(\alpha P_0(B_1) + n_1, \ldots, \alpha P_0(B_r) + n_r\right)$$

where $n_j = \#\{i : \theta_i \in B_j\}$ or $n_k = \sum_{i=1}^n \delta(A_k)$ with $\delta_i$ as point mass located at $\theta_i$ (Teh, 2010). The posterior can be rewritten as

$$P|\theta_1, \ldots, \theta_n \sim DP\left(\alpha + n, \frac{\alpha P_0 + \sum_{i=1}^n \delta_{\theta_i}}{\alpha + n}\right) = DP\left(\alpha + n, \frac{\alpha}{\alpha + n} P_0 + \frac{n}{\alpha + n} \sum_{i=1}^n \delta_{\theta_i}\right).$$

Note that the posterior base distribution in (2.2) can be interpreted as a weighted average between the prior distribution $P_0$ and empirical distribution $\sum_{i=1}^n \delta_{\theta_i}/n$. This has two important consequences. First, realisations from the DP are discrete distributions since $P \sim DP(\alpha P_0)$ implies that $P$ will be atomic having nonzero weights only on a set of atoms (Gelman et al., 2013). Secondly, as $\alpha \to 0$, the prior distribution becomes non-informative in the sense that the predictive distribution is just determined by the empirical distribution. Therefore, the empirical distribution is expected to dominate when $n$ becomes large.

In fact, the posterior base distribution given $\theta_1, \ldots, \theta_n$ is also the predictive distribution of a new observation $\theta_{n+1}$ (Teh, 2010). The predictive distribution of $\theta_{n+1}$ marginalising out
$P$ can be written as

$$p(\theta_{n+1}) = \int_P p(\theta_{n+1}, P|\theta_1, ..., \theta_n).$$

For all measurable finite set $B$, we have

$$p(\theta_{n+1} \in B|\theta_1, ..., \theta_n) = \mathbb{E}(P(B)|\theta_1, ..., \theta_n) = \frac{\alpha P_0 + \sum_{i=1}^{n} \delta_{\theta_i}(B)}{\alpha + n}.$$  

With $P$ marginalised out, we have

$$\theta_{n+1}|\theta_1, ..., \theta_n \sim \frac{\alpha P_0 + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha + n}. \quad (2.3)$$

The consequence of the predictive distribution in (2.3) is known as the Blackwell-MacQueen urn scheme, which gives a useful metaphor for interpreting Dirichlet process (Teh, 2010). We discuss the Blackwell-MacQueen urn scheme and the Chinese restaurant process in Appendix D.1 and Appendix D.2. We focus our discussion on Sethuraman’s stick-breaking construction in Section 2.5.2, which is the representation of Dirichlet process that we use for our proposed model later in Chapter 3. All three representations are important as they serve different functions. The Blackwell-MacQueen urn scheme shows the existence of the Dirichlet process, the Chinese restaurant process illustrates the key property of exchangeability, and the stick-breaking construction gives a simple and intuitive representation that provides a different avenue to posterior sampling.

### 2.5.2 Sethuraman’s stick-breaking construction

The stick-breaking construction of Sethuraman (1994) gives a direct and intuitive way to construct Dirichlet process which we use later in our proposed model of Dirichlet process misclassification mixtures model for misclassified binary data. The stick-breaking construction provides a different avenue to posterior sampling (Fruhwirth-Schnatter et al., 2019).
The stick-breaking construction derives a DP based on independent sequences of i.i.d. random variables \((v_i)_{i=1}^{\infty}\) and \((\theta^*_i)_{i=1}^{\infty}\) (where \(\theta^*_i\) denote unique component-specific parameters) and says that if

\[
v_i \overset{iid}{\sim} \text{Beta}(1, \alpha),
\]

\[
\pi_i = v_i \prod_{l=1}^{i-1} (1 - v_l),
\]

\[
\theta^*_i \overset{iid}{\sim} P_0,
\]

\[
P = \sum_{i=1}^{\infty} \pi_i \delta_{\theta^*_i},
\]

then \(P \sim DP(\alpha P_0)\) (Sethuraman, 1994). The construction can be understood metaphorically as follows. We start with a stick of length 1 and initially break off a random piece of length \(v_1 \sim \text{Beta}(1, \alpha)\). We also randomly generate \(\theta^*_1 \sim P_0\) and allocate weight \(\pi_1 = v_1\) that is the length of the stick we just broke off to it. Now length \(1 - v_1\) of the stick remains and we continue breaking off a proportion \(v_2 \sim \text{Beta}(1, \alpha)\) of the \(1 - v_1\) stick. We simulate \(\theta^*_2 \sim P_0\) and allocate the probability \(\pi_2 = v_2(1 - v_1)\) that represents the length of the stick we just broke off (Sethuraman, 1994). The process of breaking the remaining portion of the stick goes on and the stick gets shorter and shorter. Sethuraman (1994) showed that \(P = \sum_{i=1}^{\infty} \pi_i \delta_{\theta^*_i}\) defined in this way is a random probability measure distributed according to \(P \sim DP(\alpha P_0)\).

Note that by construction, the sequence \(\pi = (\pi_i)_{i=1}^{\infty}\) satisfies \(\sum_{i=1}^{\infty} \pi_i = 1\). The stick-breaking distribution of \(\pi\) is sometimes written as \(\pi \sim GEM(\alpha)\) where the letters stand for Griffiths, Engens and McCloskey (Teh, 2010). The stick-breaking representation of DP also shows that draws from a DP are discrete with probability one (Sethuraman, 1994). This discreteness makes DP well suited for the problem of placing priors on mixture components in mixture modelling. As we will see later in Section 2.6.2, with the introduction of indicator
variables specifying mixture components, the stick-breaking construction yields posterior
distribution on partitions of $\Theta$.

2.6 Applications of Dirichlet process

So far we have introduced Dirichlet process and its properties. In this section, we describe
the main applications of Dirichlet process - mixture modelling and density estimation. Be-
fore discussing Dirichlet process mixture model, we describe mixture model with Dirichlet
distribution prior and finite mixture components.

2.6.1 Mixture model with symmetric Dirichlet prior

In a finite mixture model, we assume each of $i = 1, 2, \ldots, n$ observations in the sample
belong to one of $H$ sub-population where each latent sub-population have one or more
different parameters in a parametric model. Let $z_i \in \{1, \ldots, H\}$ denote the sub-population
index for observation $i$, the response $y_i$ for observation $i$ conditioning on $z_i$ has distribution

$$y_i | z_i \sim f(\theta_{z_i}^*, \phi)$$

where $\phi$ is fixed across sub-populations and $\theta_{z_i}^*$ vary across sub-populations. Suppose the
probability of observation $i$ belong to sub-population $h$ is $p(z_i = h) = \pi_h$, we can write the
likelihood of data as

$$f(y | \pi, \theta^*, \phi) = \sum_{h=1}^{H} \pi_h f(y | \pi_h, \theta_{h}^*, \phi)$$

where $y = (y_1, \ldots, y_n)$ and $\pi = (\pi_1, \ldots, \pi_H), \theta^* = (\theta_1^*, \ldots, \theta_H^*)$.

For a finite mixture model, a Dirichlet distribution can be chosen as prior for probabilities
for each mixture component $\pi$ and $z = (z_1, \ldots, z_n)$ given $\pi$ can be assumed to follow a multi-
nominal distribution. Because of the conjugacy of Dirichlet distribution with the multinomial distribution discussed in 2.4.1, the full conditional of mixture component probabilities also follows a Dirichlet distribution.

A commonly used prior for a finite mixture model with $H$ components and mixture proportions $\pi$ is the symmetric Dirichlet prior with concentration parameter $\alpha/H$ where

$$p(\pi_1, ..., \pi_H|\alpha) \sim \text{Dirichlet}(\alpha/H, ..., \alpha/H) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/H)^H} \prod_{j=1}^{H} \pi_j^{\alpha/H-1}$$

with positive mixing proportions $\pi_j$ that sum up to 1. Given the mixing proportions $\pi = (\pi_1, ..., \pi_H)$, the cluster assignment follows multinomial distribution

$$p(z_1, ..., z_H|\pi_1, ..., \pi_H) = \prod_{j=1}^{H} \pi_j^{n_j}$$

where $n_j$ denotes the prior for occupation count of each cluster $n_j = \sum_{i=1}^{n} \delta_{\text{Kronecker}}(z_i, j)$. If $z_i = j$, observation $i$ contribute 1 to $n_j$ and no contribution otherwise. By using Dirichlet integral, we can write the conditional distribution of $z_i$ on $\alpha$ and $n_j$ where

$$p(z_1, ..., z_H|\alpha) = \int p(z_1, ..., z_H|\pi_1, ..., \pi_H)p(\pi_1, ..., \pi_H)d\pi_1, ..., d\pi_H$$

$$= \int \prod_{j=1}^{H} \pi_j^{n_j} \frac{\Gamma(\alpha)}{\Gamma(\alpha/H)^H} \prod_{j=1}^{k} \pi_j^{\alpha/H-1}d\pi_1, ..., d\pi_H = \frac{\Gamma(\alpha)}{\Gamma(\alpha/H)^H} \int \prod_{j=1}^{H} \pi_j^{n_j+\alpha/H-1}d\pi_1, ..., d\pi_H$$

$$= \frac{\Gamma(\alpha)}{\Gamma(\alpha/H)^H} B(n_j + \alpha/H) = \frac{\Gamma(\alpha)}{\Gamma(\alpha/H)^H} \frac{\prod_{j=1}^{H} \Gamma(n_j + \alpha/H)}{\Gamma(\sum_{j=1}^{H} n_j + \alpha/H)} = \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{H} \frac{\Gamma(n_j + \alpha/H)}{\Gamma(\alpha/H)}.
$$

If conjugate priors are used, Gibbs sampler can be used for posterior sampling. To sample posteriors in Gibbs sampling, we update $z_i$ conditioning on all other variables in each iteration. Therefore we need the conditional distribution of $z_i$ depending on $\alpha$ and $z_{-i}$.

87
where subscript $-i$ indicates all indexes except $i$. By (2.4), the conditional distribution of $z_i$ takes a simple form and can be written as

$$p(z_i = j | \alpha, z_{-i}) = \frac{p(z_i = j, z_1, ..., z_{i-1})}{p(z_1, ..., z_{i-1})} = \left\{ \frac{\Gamma(\alpha)}{\Gamma(n + \alpha)} \prod_{j=1}^{H} \frac{\Gamma(n_j + \alpha/H)}{\Gamma(\alpha/H)} \right\} \left/ \left\{ \frac{\Gamma(\alpha)}{\Gamma(n + \alpha - 1)} \prod_{j=1}^{H} \frac{\Gamma(n_{j,-i} + \alpha/H)}{\Gamma(\alpha/H)} \right\} \right.$$  

$$= \frac{\Gamma(n + \alpha - 1)}{\Gamma(n + \alpha)} \prod_{j=1}^{H} \frac{\Gamma(n_j + \alpha/H)}{\Gamma(n_{j,-i} + \alpha/H)} = \frac{1}{n + \alpha - 1} \prod_{j=1}^{H} \frac{n_{j,-i} + \alpha/H}{n_{j,-i} + \alpha/H}$$  

$$= \frac{n_{j,-i} + \alpha/H}{n + \alpha - 1} \quad \text{(2.5)}$$

where $n_{j,-i}$ represents the number of observations nested within the $j$th cluster excluding observation $i$ (Rasmussen, 2000). From equation (2.5), we observe that the probability of cluster assignment is invariant of the order of individual data points and total number of components $H$ is not required when sampling cluster memberships $z_i$.

If we want to have an infinite mixture model (for example in the case where we do not know the number of mixture components), one thing we can do is to use the mixture model with symmetric Dirichlet prior above and let $H$ approach infinity. When $H$ approaches infinity and each mixture component is assumed to be Gaussian, this model gives us the infinite Gaussian mixture model of Rasmussen (2000). In such case, we have

$$p(z_i = j | \alpha, z_{-i}) = \frac{n_{-i,j}}{n + \alpha - 1}$$

when $n_{-i,j} > 0$. This illustrates the clustering property of the Chinese restaurant process discussed in Section D.2 where the probability of $i$th customer sitting at table $j$ is proportional to $n_{-i,j}$, the number of other customers already sitting in each of the $j$th tables not counting the $i$th customer. The probability of $i$th observation takes up a new cluster is
since it is the complement of summing up \( \frac{n_{i,j}}{n + \alpha - 1} \) for all already occupied clusters and \( \sum_{i,j} n_{i,j} = n - 1 \). This reflects a Dirichlet Process property where the probability of a creating new cluster is proportional to the concentration parameter \( \alpha \) and a greater value of \( \alpha \) tends to encourages the creation of new clusters. In fact, as pointed out in Neal (2000), this limit of a sequence of finite mixture model where the number of mixture components \( H \to \infty \) approximates a Dirichlet process mixture model that we will discuss next.

### 2.6.2 Dirichlet process mixture model

A Dirichlet process mixture model (DPMM) is a mixture model where the mixing measure is distributed according to a Dirichlet process. It can be used to model data with unknown number of mixture components. Compared to the mixture model with large number of components \( H \), a DPMM has an unbounded \( H \) instead of a fixed \( H \). Furthermore, a Dirichlet process prior can be set such that \( \pi \) adjusts to the sparsity in the mixtures.

Let us model a set of observations \( \{x_1, \ldots, x_n\} \) using a set of latent parameters \( \{\theta_1, \ldots, \theta_n\} \). We assume each \( \theta_i \) is drawn independently and identically from \( P \) while each \( x_i \) follows distribution \( F(\theta_i) \) parameterised by \( \theta_i \). A DPMM assumes \( P \) to follow a Dirichlet process. Mathematically, we can write the DPMM model as

\[
\begin{align*}
  x_i | \theta_i & \sim F(\theta_i) \\
  \theta_i | P & \sim P \\
  P | \alpha, P_0 & \sim DP(\alpha P_0).
\end{align*}
\] (2.6)

Antoniak (1974) first proposed DPMM with the argument that DP is a desirable prior distribution. It has a large support and has analytically manageable posterior distributions given a sample of observations from the true probability distributions (Ferguson, 1973).

Since \( P \) can be represented using the stick-breaking (GEM) construction discussed in
Section 2.5.2, the factors $\theta_i$ take on value $h$ with probability $\pi_h$. We may denote this using indicator variable $z_i$ which specifies the cluster component and is distributed according to $\pi$. Therefore, we can equivalently write the DPMM as

$$
\begin{align*}
  x_i | z_i, \{\theta^*_h \} &\sim F(\theta^*_z) \\
  \theta^*_h | P_0 &\sim P_0 \\
  z_i | \pi &\sim \text{Discrete}(\pi) \\
  \pi | \alpha &\sim \text{GEM}(\alpha)
\end{align*}
$$

where $P = \sum_{l=1}^{\infty} \pi_l \delta_{\theta_l}$ and $\theta_i = \theta^*_{z_i}$. $P$ is the prior over cluster parameters and $F(\theta^*_h)$ is the distribution over the data in cluster $h$. The DPMM is an infinite mixture model with a countably infinite number of components. The number of clusters $k$, bounded by $H$, is not fixed but inferred from and grows with the data.

As stated in Section 2.6.1, the model in (2.6) and (2.7) can be expressed as the limit of a sequence of finite mixture model with symmetric Dirichlet prior where the number of mixture components $H \to \infty$. In this interpretation, the model can be written as

$$
\begin{align*}
  x_i | z_i, \theta &\sim F(\theta^*_z) \\
  z_i | \pi &\sim \text{Discrete}(\pi_1, ..., \pi_H) \\
  \theta^*_h | P_0 &\sim P_0 \\
  \pi &\sim \text{Dirichlet}(\alpha/H, ..., \alpha/H).
\end{align*}
$$

Ishwaran and Zarepour (2002) showed that by this formulation the marginal distribution induced on the observations $x_1, ..., x_n$ approaches that of a DP mixture model.
2.6.3 Density estimation

One obvious use of Dirichlet process mixture model is clustering the observations. Another common and related use is nonparametric density estimation (see Lo (1984); Escobar and West (1995); Müller et al. (2015)) since we also obtain cluster specific parameters for all components. Density estimation is concerned with inference about an unknown distribution $F$ on the basis of an observed i.i.d. sample

$$y_i | F \overset{iid}{\sim} F, \quad i = 1, \ldots, n.$$ 

In the case of Dirichlet process mixture model for density estimation, the aim is to estimate a density $p(x)$ as a DP mixture of densities

$$p(x) = \int f(x|\theta)d\pi_0(\theta)$$

where $P \sim DP(\alpha P_0)$. By the stick-breaking construction, this can be further written as

$$p(x) = \int f(x|\theta)d\pi_0(\theta) = \int f(x|\theta)d\left(\sum_{i=1}^{\infty} \pi_i \delta_{\theta_i}\right) = \sum_{i=1}^{\infty} \pi_i f(x|\theta_i^*) .$$

The empirical distribution is expected to dominate in the posterior when $n$ becomes large. This is sometimes known as the consistency property of DP where the posterior DP approaches the true underlying distribution (Teh, 2010).
2.7 Posterior inference for Dirichlet process mixture model

Posterior inference of DP mixture model is usually implemented as posterior Markov chain Monte Carlo (MCMC) simulations. In Section 2.8, we first discuss the basics of Markov chain simulation and then describe MCMC algorithms for Dirichlet process mixture models in Section 2.9.

2.8 Basics of Monte Carlo simulation

Let \( \theta \) denote parameters of interest, \( y \) denote observed data, Bayesian inference aims to make probability statements about posterior density \( f(\theta|y) \) (Gelman et al., 2013). The \( f(\theta|y) \) term is often rewritten via Bayes’ rule where

\[
f(\theta|y) = \frac{f(\theta)f(y|\theta)}{f(y)} \propto f(\theta)f(y|\theta).
\]

We would like to compute \( f(\theta|y) \) but it is usually intractable. The high level idea of what we would then do is to approximate the intractable distribution via sampling, i.e. sample a set of draws from that distribution and consider the empirical distribution of these samples as the approximation to the desired distribution. This allows us to convert any expectations expressed as integrals over the intractable posterior distribution as finite sums over the empirical distribution approximation.

Markov chain Monte Carlo is a general method based on drawing values of \( \theta \) from approximate distributions and then correct those draws to better approximate the target posterior distribution \( p(\theta|y) \) (Gelman et al., 2013). A Markov chain is a sequence of random variables \( \theta^{(1)}, \theta^{(2)}, \ldots \) where for any \( t \) the distribution of \( \theta^{(t)} \) given all previous \( \theta \)’s depends only the
most recent value $\theta^{(t-1)}$. The key is that approximate distributions are improved at each step in the simulation in the sense of converging to the target distribution (Gelman et al., 2013).

Monte Carlo simulation is often used when it is not possible or computationally efficient to sample $\theta$ directly from $p(\theta|y)$. Instead, we sample iteratively such that at each step we expect to draw from a distribution that is closer to $p(\theta|y)$. The samples form a sequence $\theta^{(1)}, \theta^{(2)}, \ldots$ where $\theta^{(t)}$ is drawn depending on previous draw $\theta^{(t-1)}$ from a transition distribution $T_t(\theta^{(t)}|\theta^{(t-1)})$. The transition probability distributions must be constructed so that the Markov chain converges to a unique stationary distribution that is the posterior distribution $p(\theta|y)$ (Gelman et al., 2013). Often times, conjugate distributions are used as priors so that the posterior conditional distributions are easy to sample from.

Next, we introduce the basic Markov chain simulation methods - the Gibbs sampler and the Metropolis-Hastings algorithm.

### 2.8.1 Gibbs sampler

The Gibbs sampler, or alternating conditional sampling, divides parameter vector $\theta$ into $d$ subvectors $\theta = (\theta_1, \ldots, \theta_d)$ and draws each subset conditional on the values of all others (Gelman et al., 2013). At iteration $t$, an ordering of the $d$ subvectors of $\theta$ is chosen and $\theta_j^{(t)}$ is sampled from the conditional distribution given all other components $p(\theta_j^{(t+1)}|\theta_{-j}^{(t)}, y)$ where $\theta_{-j}^{(t)}$ represents the current values of all the components of $\theta$ except component $j$. It can be shown that the resulting samples will be asymptotically draws from the desired posterior distribution.
2.8.2 Metropolis and Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm is an adaptation of a random walk with an acceptance/rejection rule to ensure that the chain converges to the specified target distribution. The algorithm proceeds with a proposal distribution (or jumping distribution) that is easy to sample from.

For $t = 1, 2, \ldots$ :

1. Sample a proposal $\theta^*$ from the proposal distribution at time $t$, $q(\theta^*|\theta^{(t-1)})$.

2. Calculate the ratio of the densities in Section

$$r = \frac{p(\theta^*|y)/q(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)/q(\theta^{(t-1)}|\theta^*)}.$$

3. Set

$$\theta^{(t)} = \begin{cases} 
\theta^* & \text{with probability } \min(r, 1) \\
\theta^{(t-1)} & \text{otherwise.}
\end{cases}$$

The Metropolis algorithm is the special case where the proposal distribution is symmetric and ratio simplifies to $r = \frac{p(\theta^*|y)}{p(\theta^{(t-1)}|y)}$. The Gibbs sampler is the special case where the proposal distribution only jumps at $j$th subvector at step $j$ of iteration $t$ and ratio $r$ always equals to 1. The basic idea of Gibbs sampling is to iteratively sample from conditional distribution $p(\theta_d|y, \theta_{-d})$ where $\theta_{-d}$ is $\theta$ without the $d$th parameter. $p(\theta|y)$ can be rewritten as $p(\theta|y) = p(\theta_d, \theta_{-d}|y) = p(\theta_d|y, \theta_{-d})p(\theta_{-d}|y)$. Therefore,

$$r = \frac{p(\theta^*|y)q(\theta^{(t-1)}|\theta^*)}{p(\theta^{(t-1)}|y)q(\theta^*|\theta^{(t-1)})} = \frac{p(\theta^*|y, \theta^*_{-d})p(\theta^*_{-d}|y)p(\theta_d|y, \theta_{-d})}{p(\theta_d|y, \theta_{-d})p(\theta_{-d}|y)p(\theta^*_d|y, \theta^*_{-d})} = 1$$

as $\theta^*_{-d} = \theta_{-d}$ and sampling a proposal $\theta^*$ just means sample from $p(\theta^*_d|y, \theta^*_{-d})$. 

94
2.9 Markov chain Monte Carlo methods for Dirichlet process mixture model

In this section, we give a brief overview of Markov chain Monte Carlo methods for Dirichlet process mixture model. We focus our main discussion on Gibbs samplers for conjugate model which is the case of our proposed method for inference for binary data in the presence of label misclassification in Section 3.

Broadly speaking, Markov chain Monte Carlo methods for Dirichlet process mixture model can be roughly categorised into marginal and conditional method (Papaspiliopoulos and Roberts, 2008). Marginal methods exploit convenient mathematical properties of the Dirichlet process and analytically integrate out the infinite-dimensional $P$. Samples from the marginal distribution of the remaining variables can be obtained using the Gibbs sampler for conjugate models where further integrations can be performed analytically (Papaspiliopoulos and Roberts, 2008). We do not use marginalised Gibbs sampler for our model but include a summary of it in Appendix D.3.

Conditional methods impute the Dirichlet process and update it as a component of the Gibbs sampler. Since this requires imputation of an infinite-dimensional process, implementation of the conditional methods relies on some approximation (Papaspiliopoulos and Roberts, 2008). In Section 2.9.1, we discuss in detail the method of Ishwaran and James (2001) which makes finite approximations of a Dirichlet process.

When the sampling model and base measure of DP prior are not conjugate, Gibbs sampler cannot be used but other more complex schemes such as Metropolis Hastings like algorithm exist and can be found in Neal (2000). There are also other alternatives using slice samplers and retrospective samplers. Slice samplers for Dirichlet process mixture models can be found in Walker (2007), Kalli et al. (2011) and retrospective samplers can be found in
Papaspiliopoulos and Roberts (2008) and Hastie et al. (2015). Furthermore, there are alternative procedures to the marginal and conditional approaches, for instance, variational inference procedure proposed by Blei et al. (2006) and a sequential greedy search algorithm for selecting the partition for clusters by Wang and Dunson (2011).

### 2.9.1 Truncated Gibbs sampler

The *truncated Gibbs sampler*, also called *Blocked Gibbs sampler* by Gelman et al. (2013), is a Gibbs sampler proposed by Ishwaran and James (2001) based on approximation of the DPMM with $H$ number of components. Ishwaran and James (2001) give closed form for the moments for two-parameter Poisson-Dirichlet (Pitman-Yor) process $\mathcal{PY}(a, b)$, which Dirichlet process is a special case of with $a = 0, b = \alpha$. This moment simplifies further in the case of Dirichlet process and decreases exponentially in components size $H$ (Ishwaran and James, 2001). Therefore, with large enough $H$, an accurate approximation of the infinite case Dirichlet process can be achieved with a finite Dirichlet process like the case of symmetric Dirichlet prior where $H \to \infty$ described in Section 2.6.1.

Ishwaran and James (2001) describe the sampling steps of the truncated Gibbs sampler that effectively sample $P \sim DP_H$. With a pre-determined upper bound $K$ for the number of components where $H \leq K$, Escobar and West (1995) and Ishwaran and Zarepour (2002) propose a blocked Gibbs sampler with the following steps:

1. update $z_i \in \{1, \ldots, K\}$ by multinomial sampling with $p(z_i = h|\theta^*, \pi, x_i) = \frac{\pi_h f(x_i|\theta^*_h)}{\sum_{h=1}^{K} \pi_f f(x_i|\theta^*_f)}$ for $h = 1, \ldots, K$.

2. update the stick-breaking weight $v_h \sim Beta \left(1 + n_h, \alpha + \sum_{s=h+1}^{K} n_s\right)$ for $h = 1, \ldots, K - 1$ where $n_h$ denotes the number of observations in cluster $h$; set $v_K = 1$ to ensure $\pi_j = 0$ for all $j = K + 1, \ldots \infty$.

3. use the results of $v_1, \ldots, v_K$ to update $\pi_1, \ldots, \pi_K$. 


4. update $\theta^*_h$ for $h = 1, ..., K$ from its posterior for $n_h > 0$ or sample $\theta_h$ from prior $P_0$ if $n_h = 0$.

This algorithm of Ishwaran and James (2001) is also sometimes called the *Accelerated Pólya Urn Gibbs sampler* since it accelerates the Pólya Urn Gibbs sampler by updating parameters in blocks instead of one at a time like in the Pólya Urn Gibbs sampler.

Note that to see the posterior of $v$ (vector of Beta distributed random variables that helps defining mixture component probabilities) in the truncated stick-breaking process, recall that to arrive at $P \sim DP(\alpha P_0)$, we have independent sequences of i.i.d. random variables $(v_i)_{i=1}^\infty$ and $(\theta^*_i)_{i=1}^\infty$ where

\begin{align*}
v_i &\sim \text{Beta}(1, \alpha) \\
\pi_i &= v_i \prod_{l=1}^{i-1} (1 - v_l) \\
\theta^*_i &\sim P_0 \\
P &= \sum_{i=1}^\infty \pi_i \delta_{\theta^*_i}.
\end{align*}

For the posterior of $v_h$, $\pi_h$ for the stick-breaking process, we have

\begin{align*}
p(v_h|\pi, z) &= p(v_h)p(z|\pi, v)p(\pi|v) \\
&= \frac{(1 - v_h)^{\alpha - 1}}{B(1, \alpha)} \frac{n!}{n_1!n_2!...} \times \prod_{i=1}^\infty \pi_i^{n_i} \\
&\propto (1 - v_h)^{\alpha - 1} \times \prod_{i=1}^{\infty} \left( v_i \prod_{s=1}^{i-1} (1 - v_s) \right)^{n_i} \\
&\propto v_h^{n_h}(1 - v_h)^{\sum_{s > h} n_s + \alpha - 1}
\end{align*}

since only terms with $v_s$ matter. With the truncated Normal, the index $s$ only goes up to
Therefore, in posterior, we have

\[ v_h \sim \text{Beta}\left( 1 + n_h, \alpha + \sum_{s=h+1}^{K} n_s \right) \text{ for } h = 1, ..., K - 1 \]

\[ v_K = 1 \] to ensure \( \pi_j = 0 \) for all \( j = K + 1, \ldots, \infty \)

\[ \pi_i = v_i \prod_{t=1}^{i-1} (1 - v_t). \]

### 2.9.2 Hyperprior distribution

In choosing the priors for the DP, choices need to be made regarding \( P_0 \) and \( \alpha \). \( P_0 \) can be thought as inducing the prior for the cluster locations and the common practice is to standardise the data \( X \) and choose \( P_0 \) to be centered near zero with unit variance (Gelman et al., 2013). The precision parameter \( \alpha \) plays a key role in controlling the prior on the number of clusters and can be either set to a small value such as \( \alpha = 1 \) or assigned a hyperprior. Setting \( \alpha \) to a small value favours allocation to few clusters relative to sample size and using hyperprior allow the value of \( \alpha \) to be informed by the data (Gelman et al., 2013).

A common choice of hyperprior for \( \alpha \) is \( \alpha \sim \text{Gamma}(a, b) \). For the stick breaking truncation with at most \( K \) components, we have

\[
p(\alpha | \cdot) \propto \alpha^K \prod_{h=1}^{K} (1 - v_h)^{a} \times \alpha^{a_0 - 1} \exp(-b_0 \alpha) \\
\quad \times \alpha^{a_0 + K - 1} \exp \left( - \left[ b_0 - \sum_{h=1}^{K} \log(1 - v_h) \right] \alpha \right),
\]

which is a Gamma distribution. To help improve mixing, when the truncated level is sufficiently large (which is the case of our experiment in Section 4), we could instead update based on the marginal sampler described below.
As shown in West (1992), the conditional posterior of $\alpha$ is only based on previously sampled value of the number of clusters $k$ and has posterior

$$p(\alpha | k) \propto p(\alpha) p(k | \alpha) = p(\alpha) \alpha^{k-1} (\alpha + n) B(\alpha + 1, n)$$

$$= p(\alpha) \alpha^{k-1} (\alpha + n) \int_0^1 \eta^n (1 - \eta)^{n-1} d\eta$$

where $n$ is the sample size. This implies that $p(\alpha | k)$ is the marginal distribution from a joint for $\alpha$ and continuous quantity $\eta \in [0, 1]$ such that

$$p(\alpha, \eta | k) \propto p(\alpha) p(k | \alpha, \eta) = p(\alpha) \alpha^{k-1} (\alpha + n) \eta^n (1 - \eta)^{n-1},$$

where $\alpha > 0$ and $0 < \eta < 1$ (West, 1992). Therefore, a Gamma prior for $\alpha$ where $\alpha \sim Gamma(a_{\alpha}, b_{\alpha})$ leads to a convenient posterior

$$p(\alpha | \eta, k) \approx \alpha^{a_{\alpha} + k} (\alpha + n) e^{-\alpha (b_{\alpha} - \log(\eta))}$$

$$\approx \alpha^{a_{\alpha} + k - 1} e^{-\alpha (b_{\alpha} - \log(\eta))} + n \alpha^{a_{\alpha} + k - 2} e^{-\alpha (b_{\alpha} - \log(\eta))}.$$ For $\alpha > 0$, this posterior of $\alpha$ reduces to a mixture of two Gamma distributions where

$$\alpha | \eta, k \sim \pi_{\eta} Gamma(a_{\alpha} + k, b_{\alpha} - \log(\eta)) + (1 - \pi_{\eta}) Gamma(a_{\alpha} + k - 1, b_{\alpha} - \log(\eta))$$

where $p(\eta | \alpha, k) \propto \eta^a (1 - \eta)^{n-1}$, therefore $\eta | \alpha, k \sim Beta(\alpha + 1, n)$ (West, 1992). The weights for the two Gamma’s is specified as $\pi_{\eta}$ and $1 - \pi_{\eta}$ where $\pi_{\eta}$ is defined by

$$\frac{\pi_{\eta}}{1 - \pi_{\eta}} = \frac{a_{\alpha} + k - 1}{n (b_{\alpha} - \log(\eta))}.$$
Rearranging the above equation leads to

\[ \pi_\eta = \frac{a_\eta + k - 1}{n(b_\eta - \log(\eta)) + (a_\eta + k - 1)}. \]

By further introducing a uniform random variable \( u \sim U[0, 1] \) we alternatively sample \( \alpha \) from its posterior following the procedures

\[
\eta \sim Beta(\alpha + 1, n), \quad u \sim U[0, 1] \\
b = b_\alpha - \log(\eta), \quad I = \mathbb{I}\left[ \frac{u}{1 - u} \geq \frac{a_\alpha + k - 1}{nb} \right] \\
\alpha \sim Gamma(a_\alpha + k - I, b). \tag{2.10}
\]

Equation (2.10) can be derived from the probability of the indicator, which takes the value of

\[
p\left[ \frac{u}{1 - u} \geq \frac{a_\alpha + k - 1}{nb} \right] = p\left[ u \geq \frac{a_\alpha + k - 1}{nb + (a_\alpha + k - 1)} \right] \\
= 1 - \frac{a_\alpha + k - 1}{nb + (a_\alpha + k - 1)} = \frac{nb}{nb + (a_\alpha + k - 1)}. 
\]

For \( u \sim U[0, 1] \), the probability of \( p(u < s) = 1 - s \) for \( 0 \leq s \leq 1 \). In our case, this probability is exactly \( 1 - \pi_\eta \), thus (2.10) is equivalent to sampling \( Gamma(a_\alpha + k - 1, b_\alpha - \log(\eta)) \) with probability \( 1 - \pi_\eta \) and sampling \( Gamma(a_\alpha + k, b_\alpha - \log(\eta)) \) with probability \( \pi_\eta \).
Chapter 3

Dirichlet process probit misclassification mixture model for misclassified binary data

In previous chapters, we introduced Dirichlet process and Dirichlet process mixture models which can perform density estimation and clustering on data with unknown or unspecified number of clusters. In this chapter, we will introduce our proposed Bayesian mixture model for drawing inferences about contaminated binary data based on joint Dirichlet process mixture model of covariates $X$ and observed responses $y$. We later refer to our model as Dirichlet process probit misclassification mixture model (DP-PMM).

DP-PMM is based on Dirichlet process mixtures, which probabilistically clusters observations into groups with similar covariates and similar misclassification probabilities. DP-PMM is motivated by real-life scenarios where test accuracy is a function of some covariates. For example, in study of Covid-19 RT-qPCR tests, false-negative rates are found to be strongly associated with age and sex and in fecal immunochemical tests for colorectal cancer screening, sensitivity are lower in higher age groups (Levine-Tiefenbrun et al., 2020; Selby et al.,...
The Bayesian approach of DP-PMM enables nonparametric estimation of differential misclassification probabilities. Not only it resolves potential identifiability issues in the likelihood of the observed data mentioned in Section 1.1, DP-PMM also allows easy incorporation of information about specificity and specificity of observed responses if available.

The main idea of DP-PMM can be summarised as follows. We assume that the covariates have latent structure which is also associated with heterogeneity in sensitivity and specificity across components. The observations are partitioned into various clusters based on $X$ and $y$ where the potential misclassification in $y$ are incorporated via priors on cluster-specific misclassification probabilities $\gamma_0, \gamma_1$. $\gamma_0 = (\gamma_{01}, \gamma_{02}, \ldots)$ and $\gamma_1 = (\gamma_{11}, \gamma_{12}, \ldots)$ are vectors of probability of misclassification in different clusters when the true response is 0 or 1. Since they are cluster-specific, the learned misclassification probabilities are thus differential across observations. The misclassification probabilities can be regularised via their priors if additional information like sensitivity and specificity of the data are available. The posterior distribution of DP-PMM contains both the binary regression parameters, as well as observation-specific misclassification probabilities.

### 3.1 Model structure

DP-PMM is a Dirichlet process mixture model of $X$ and $y$. The assumptions are that $X$ can be approximated by a Dirichlet mixture of multivariate Gaussians with means $\mu$ and covariance matrices $\Sigma$ and $y$ reflect different misclassification probabilities across the clusters. The true responses are $t = (t_1, \ldots, t_n)$ and $t_i$ is assumed to be a probit regression model of $x_i$ parametrised by $\beta$ where $\eta_i = \Phi(x_i \beta)$. Observed response $y_i$ is assumed to be a contaminated version of $t_i$ whose misclassification probability is determined by $\gamma_0, \gamma_1$. Given the cluster membership, the misclassification probabilities are fixed where $\gamma_{0i} = \gamma_{0h}$, $\gamma_{1i} = \gamma_{1h}$ for all observation $i$ that are in cluster $h$. Using $K$ to denote the total number of components.
subscript $h$ to denote the cluster index and $\pi = (\pi_1, \ldots, \pi_K)$ to denote cluster membership probabilities, our joint Dirichlet process mixture model assumption can be summarised as

$$f(X, y | \Theta^*, \beta) = \sum_{h=1}^{K} \pi_h f(X, y | \theta^*_h, \beta) = \sum_{h=1}^{K} \pi_h f(X | \theta^*_h) f(y | X, \theta^*_h, \beta)$$ (3.1)$$

where $\theta^*_h = (\mu^*_h, \Sigma^*_h, \gamma^*_h)$ represents cluster-specific parameters for cluster $h$ and $\gamma^*_h = (\gamma^*_0h, \gamma^*_1h)$. $X$ is determined by the Gaussian components and $y$ is determined by the true responses $t$ and misclassification probabilities $\gamma^*_0h, \gamma^*_1h$. Consequently, (3.1) can be further written as $f(X, y | \Theta^*, \beta) = \sum_{h=1}^{K} \pi_h f(X | \mu^*_h, \Sigma^*_h) f(y | X, \gamma^*_0h, \gamma^*_1h, \beta)$.

We introduce discrete random variable $z$ to denote the cluster membership of observations where $z_i = h$ implies that observation $i$ is in cluster $h$. In the joint Dirichlet process mixture model, the clustering is not just dependent on $X$ but also takes into consideration of potential misclassification in $y$. $X$ is a mixture of Gaussians where its mixture components follow a Dirichlet process with concentration parameter $\alpha$. Mathematically, the model assumptions can be written in a hierarchical model

$$p(y_i = s | t_i = s, z_i = h, \gamma_0, \gamma_1) = 1 - \gamma^*_{sh} \quad \text{where } s = \{0, 1\}$$

$$t_i | x_i, \beta \sim \Phi(x_i, \beta)$$

$$x_i | \mu^*_h, \Sigma^*_h, z_i = h \sim F(\mu^*_h, \Sigma^*_h)$$

$$\mu_i | \mu^*_h, \Sigma^*_h, \gamma_{0h}, \gamma_{1h} \sim P$$

$$P | \alpha, P_0 \sim DP(\alpha P_0)$$

$$\alpha \sim \pi(\alpha)$$

(3.2)

where $\pi(\alpha)$ denotes the prior for $\alpha$ and $F(\mu^*_h, \Sigma^*_h)$ represents the distribution parametrised by $\mu^*_h, \Sigma^*_h$ that $x_i$ is generated from. In general, we use subscript $i$ to denote variation across observations and subscript $h$ to denote variation across clusters.
Graphically, we can visualise DP-PMM using Figure 3.1 where shaded circles denote observed variables and unshaded circles denote unobserved variables and parameters. We observe contaminated response $y$ and covariates $X$ where there are $n$ observations in total. The number of clusters is assumed to be $K$ at max, which allows finite approximation of the DP with infinite number of clusters that yield to a simplified sampling algorithm that we describe in Section 3.4.

Figure 3.1: Graphical model representation of data generating process (shaded circles denote observed variables and unshaded circles denote unobserved variables and parameters); $\theta^* = (\mu^*, \Sigma^*, \gamma^*)$ represents cluster-specific Gaussian mean, variance, and misclassification probabilities. $n$ denotes the size of the data and $K$ represents the maximum number of clusters in the data.

The ultimate interest of DP-PMM is to obtain posterior inference of $\beta, \theta^*, z, t$ where $\beta$ parametrises the probit regression model for the true responses and $\gamma_0^*, \gamma_1^*$ and $z$ informs us about the misclassification in the observed data. The steps of DP-PMM are given in details
in Section 3.4 and are motivated by the following decomposition where

\[
p(\beta, \gamma_0^*, \gamma_1^*, \mu^*, \Sigma^*, \pi, z, t, \alpha|X, y) \propto p(y|\beta, \gamma_0^*, \gamma_1^*, \mu^*, \Sigma^*, \pi, z, t, X)p(\beta, \gamma_0^*, \gamma_1^*, \mu^*, \Sigma^*, \pi, t, \alpha|X)
\]

\[
= p(y|t, \gamma_0^*, \gamma_1^*, z)p(\gamma_0^*, \gamma_1^*)p(\mu^*, \Sigma^*, \pi, z, t, \alpha|X)
\]

\[
= p(y|t, \gamma_0^*, \gamma_1^*, z)p(\gamma_0^*, \gamma_1^*)p(z|\pi)p(t|\beta, X)p(\pi, \alpha, \mu^*, \Sigma^*, \beta, X)
\]

\[
= p(y|t, \gamma_0^*, \gamma_1^*, z)p(\gamma_0^*)p(\gamma_1^*) \times p(z|\pi)p(\pi|\alpha)p(\alpha)
\]

\[
\times p(t|\beta, X)p(\beta) \times p(X|\mu^*, \Sigma^*, \pi(\mu^*)p(\Sigma^*).
\]

\[(3.3)\]

The first line is just an application of Bayes rule. The second line simplifies because \(y\) only depends on \(\gamma_0^*, \gamma_1^*, t, z\) and \(\gamma_0^*, \gamma_1^*\) is independent of \(X\). The third and forth line rely on the fact that \(t\) only depends on \(X, \beta\) and \(z\) is determined by \(\pi\) through the influence of concentration parameter \(\alpha\) and Gaussian component parameters \(\mu^*, \Sigma^*\). The \(p(t|\beta, X)p(\beta)\) in equation (3.3) show how we learn \(\beta\) conditional on \(t\) and the \(p(y|t, \gamma_0^*, \gamma_1^*, z)p(\gamma_0^*)p(\gamma_1^*)\) are how we learn \(\gamma_0^*, \gamma_1^*\) on \(z\).

### 3.2 Model priors

For all clusters, the prior assumptions for the Gaussian component \(h\) are

\[
\Sigma_h^* \sim W^{-1}(m_h, \Psi_h),
\]

\[
\mu_h^*|\Sigma_h^* \sim MVN(\epsilon_h, \Sigma_h^*/k_h)
\]

\[(3.4)\]

where \(m_h, \Psi_h, \Sigma_h^*, \epsilon_h, k_h\) could be component specific or all the same for different components. The latent class membership \(z\) takes discrete values 1, 2, ... and is assumed to follow a discrete
distribution parametrised by probability vector $\pi$ where

$$z|\pi \sim \text{discrete}(\pi).$$

We do not know nor specify the number of clusters in DP-PMM and assume $\pi$ to follow a Dirichlet process prior, which can be represented using the stick breaking construction where

$$\pi \sim GEM(\alpha).$$

The stick breaking construction of $\pi$ implies that it can be obtained by independent and identically distributed (i.i.d.) sequence of auxiliary random variable $v_l$ where

$$\pi_s = v_s \sum_{l=1}^{s-1} (1 - v_l),$$

$$v_l \overset{iid}{\sim} \text{Beta}(1, \alpha).$$

Here $\alpha$ is the concentration parameter and plays a key role in controlling the prior on the number of clusters (Gelman et al., 2013). If $\alpha$ is fixed at a small value like 1, the allocation will favour few clusters relative to the size of the data (Gelman et al., 2013). In DP-PMM, we want to allow the data to inform about the appropriate value of $\alpha$ thus we assign a hyperprior for $\alpha$. For conjugacy, we assume $\alpha$ follows a Gamma distribution where

$$\alpha \sim \text{Gamma}(a_\alpha, b_\alpha).$$

The true response $t_i$ is assumed to be modeled by a probit regression model of $x_i$ with link function $g^{-1} = \Phi$ and parameters $\beta$ such that for observation $i$, $p(t_i = 1) = \Phi(x_i \beta)$.  

106
We assume a multivariate Gaussian prior for $\beta$ where

$$\beta \sim MVN(m_0, V_0).$$

$y_i$ is linked to $t_i$ by the latent classes $z_i$ and misclassification probabilities $\gamma^*_0, \gamma^*_1$. By definition, the probability that the $i$th observation is misclassified is $\gamma^*_0$ when $t_i = 0$ and $\gamma^*_1$ when $t_i = 1$ given $z_i = h$. Our assumption of cluster-specific misclassification probabilities implies that given the cluster, the misclassification probabilities are non-differential (i.e. fixed) across observations in that cluster. We assume the same Beta prior for all $\gamma^*_0h$ and another Beta prior for all $\gamma^*_1h$ regardless of cluster index $h$. Mathematically, we can express the priors for misclassifying observation $i$ as

$$\gamma^*_0h \sim Beta(a_0, b_0), \quad \gamma^*_1h \sim Beta(a_1, b_1).$$

where $z_i = h$ taking discrete values in $\{1, 2, ..., K\}$ with $K$ as the maximum number of clusters.

### 3.3 Simple example

We have yet to describe how to do posterior inference for DP-PMM in Section 3.4. To demonstrate how DP-PMM provide inference, we apply DP-PMM to a simulated dataset and illustrate some of the outputs.

#### 3.3.1 Simulated data

We simulate $X$ from a mixture of three multivariate Gaussians $MVN(\mu_1, \Sigma_0), MVN(\mu_2, \Sigma_0), MVN(\mu_3, \Sigma_0)$ where

$$\mu_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mu_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mu_3 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and $\pi = (0.5, 0.3, 0.2)$. 

107
The true responses are simulated from a generalised linear model with probit link where
\[ p(t_i = 1) = g^{-1}(x_i \beta) \] and \( \beta = (0, 2, -1)^T \). We split the data into training and test data.

For training data, we generate observed responses by using the same latent structure for the Gaussians mixture model and the misclassification. In other words, \( y \) is generated from \( t \) using \( (\gamma_{01}, \gamma_{11}) = (0.1, 0), (\gamma_{02}, \gamma_{12}) = (0.05, 0.05), \) and \( (\gamma_{03}, \gamma_{13}) = (0, 0.1) \).

A visualisation of the observed responses in comparison to the true responses are shown in Figure 3.2.

Figure 3.2: Visualisation of the true responses (left) and observed contaminated responses (right) in training data where the green line indicates the decision boundary for the true responses.

We apply DP-PMM to the observed simulated dataset with contaminated responses assuming that there are 30 components and all components have \( \gamma_{00}, \gamma_{11} \) following prior Beta\((1, 20)\). We further assume the prior for regression coefficients \( \beta \) for the probit model for true responses is a multivariate Gaussian with mean \( 0 \) and covariance matrix a \( 3 \times 3 \) matrix with 9’s on the diagonals. Following the steps described in Algorithm 8, we run DP-PMM for 5000 MCMC iterations after burn-in.
3.3.2 DP-PMM outputs

DP-PMM produces many outputs which includes the cluster assignments of the observations. The cluster membership of observations depends on the covariates as well as the misclassification in the observed response. To summarise the cluster assignments from all iterations, we use the optimal point estimate that minimises the posterior expectation of variational information proposed by Wade and Ghahramani (2018). In this point estimate, variational information is used as the loss function because it is possess the desirable property of being a metric on the space of partitions (thus invariance to label switching) and compares the information in clusterings with the information shared among clusterings (Wade and Ghahramani, 2018). In Figure 3.3, we visualise the best partition of cluster assignments in comparison to the true GMM cluster membership of the training data. As we can see, the two cluster membership plots are very similar indicating that the structure of the covariates is picked up by the method.

![Figure 3.3: Visualisation true membership of observations to the Gaussian components (left) and the best cluster assignment (right) in simulated dataset.](image)

Besides the best point estimate that minimises the posterior expectation of variational information, the clustering result can also be visualised by heat map of posterior similarity matrix of training data as shown in Figure 3.4. For a dataset of size $n$, the posterior similarity
matrix is a $n \times n$ matrix that represent the probabilities that two data points are in the same cluster, estimated from the proportion of MCMC samples that cluster the two data points together (Wade and Ghahramani, 2018). The posterior similarity matrix in Figure 3.4 is plotted by R package \textit{mcclust.ext} of Wade and Wade (2015).

![Figure 3.4: Heat map of posterior similarity matrix of clustering on training data of simulated dataset.](image)

As of misclassification, DP-PMM also output the misclassification probabilities of each cluster in each iteration. In Figure 3.5, we plot the histograms of average $\gamma_{0h}^*$ and $\gamma_{1h}^*$ in all nonempty clusters. As we can see in this plot, there are many clusters with average misclassification probabilities $\gamma_{0h}^*$ and $\gamma_{1h}^*$ close to 0, 0.05, or 0.1, which are the true misclassification probabilities in three clusters that generates the data. Note that for this simulated dataset, we assumed 30 components and not all components need to be occupied. In fact, if we check the number of clusters (nonempty components) in all iterations as shown in the right plot of Figure 3.6, we see that the number of clusters are always less than 20 and most of time, less than 10. This confirms that 30 is an appropriate component number to assume for this dataset.
Figure 3.5: Histogram of average $\gamma_{0h}^*$ in nonempty clusters (left) and histogram of average $\gamma_{1h}^*$ in nonempty clusters (right).

One of the main motivations of DP-PMM is to directly provide inference for true responses while only observing contaminated responses. An important output of the model is the distribution of estimated regression coefficients for the probit model assumed for true responses. In the left of Figure 3.6, we visualise the distribution of estimated regression coefficients for the probit model. As we can see in the plot, the 95% credible intervals all contain the corresponding coefficient of the true probit regression model.

Figure 3.6: Visualisation of model output: estimated regression coefficients for probit model with vertical lines indicating the 95% credible interval (left) and the number of nonempty clusters in all iterations (right).
3.3.3 Prediction using DP-PMM

If the interest is to predict unseen observation, we can utilise the posterior distributions of estimated probit regression coefficients. For a new observation $x_{n+1}$, each set of probit regression coefficients in iteration $(r)$ gives an estimate of the probability that the true response is equal to one, $p(t_{n+1} = 1|x_{n+1}, \hat{\beta}^{(r)})$. To incorporate the uncertainty in the estimated regression coefficients, we can use the average probability that the true response is equal to one, obtained from probit regression coefficients from all MCMC iterations where

$$p(t_{n+1} = 1|x_{n+1}, \text{the data}) \approx \frac{1}{R} \sum_{r=1}^{R} \Phi(x_{n+1}\hat{\beta}^{(r)}) .$$

We visualise the average probability that the true response is equal to one for unseen test data from all 5000 MCMC iterations in Figure 3.7. As we can see in the plot, the average probability that the true response is equal to one for unseen test data seem to be in line with the predictions from the true discriminative line learned from the true responses of training data.

![Figure 3.7: True responses of test data (left) average probability of 1 for test data using estimated probit regression coefficients from all 5000 MCMC iterations (right).](image)

With predictive probabilities for each observation in the the test set, we can also compute standard classification metrics like classification accuracy, false positive true, false negative
rate and etc.. We compute these metrics for the experiment study with a real dataset in Section 4.

3.4 Posterior inference for proposed model

In this section, we describe a Gibbs-like procedure to obtain the posterior inference for DP-PMM motivated by the truncated Gibbs sampler of Ishwaran and James (2001). Gibbs sampling is possible in DP-PMM because we have chosen conjugate priors for our base measure (in this case, parameters within clusters $\mu^*, \Sigma^*, \gamma_0^*, \gamma_1^*$) and mixing proportions $\pi$. We use a stick-breaking prior for $\pi$ that assumes infinite number of components in theory and for computational convenience approximate it with a pre-determined maximum number of components $K$. In a real dataset, the number of components has to be finite anyway since the number of components cannot be larger than the sample size. Next, we derive conditional posterior distributions used in our Gibbs sampler.

3.4.1 Posterior of cluster-specific parameters $\theta$

For the first part of our Gibbs sampler, we update component specific parameters $\mu^*, \Sigma^*, \gamma_0^*, \gamma_1^*$ for component 1, 2, ..., $K$. These parameters are fixed conditioning on latent cluster assignment $z$ and covariates $X$ where we can write $p(\theta^*|z) = p(\theta^*|z, X)$. In iteration $t$, we update the Gaussian component parameters for component $h$ using observations that satisfy $z_i^{(t-1)} = h$. Specifically, we have

$$
\Sigma_h^{(t)}|X, z^{(t-1)} \sim W^{-1}\left(n_h + n_h^{(t-1)}, \Psi_h + \Lambda_h^{(t-1)} + \frac{n_h^{(t-1)}}{n_h^{(t-1)} + k_h}(\bar{x}_h - \epsilon_h)^T(\bar{x}_h - \epsilon_h)\right)
$$

(3.5)

where $\Lambda_h^{(t-1)} = \sum_{i: z_i = h}(x_i - \bar{x}_h)(x_i - \bar{x}_h)^T$ and $\bar{x}_h$ the average value of $x_i$ of observations in cluster $h$. The scale matrix is shaped by the prior opinion of $\Sigma_h$, namely $\Psi_h$, the sum of
squares $\Lambda_h^{(t-1)}$ in cluster $h$ in previous iteration $t-1$, and the deviation between prior and estimated mean values (Franzén, 2006). Given $\Sigma_h^{* (t)}$, the posterior distribution for Gaussian mean on iteration $t$ is also conjugate, following

$$
\mu_h^{* (t)} | X, \Sigma_h^{* (t)}, \mathbf{z}^{(t-1)} \sim MVN \left( \frac{\bar{\mathbf{e}}_h^{(t-1)}}{n_h^{(t-1)} + k_h}, \frac{\Sigma_h^{* (t)}}{n_h^{(t-1)} + k_h} \right)
$$

where $\bar{\mathbf{e}}_h^{(t-1)} = \frac{\Psi_{ik_h} + n_h^{(t-1)} \bar{z}_h}{n_h^{(t-1)} + k_h}$ is the weighted sum of the prior and estimated mean values from the data from previous iteration.

We perform these updates of Gaussian component parameters for $h = 1, 2, \ldots, k$. If a cluster $h$ is empty, i.e. $\# \{z_i = h\} = 0$, we sample $\Sigma_h^{* (t)}, \mu_h^{* (t)}$ from its prior in (3.4) instead of its posterior in (3.5) and (3.6).

Besides the mean and covariance of the Gaussian components, the misclassification probabilities are also cluster-specific parameters, which we update next based on previous round values of cluster membership $\mathbf{z}^{(t-1)}$. For cluster $h$, the updates of $\gamma_{0h}^*, \gamma_{1h}^*$ is based on $p(\gamma_{0h}^*, \gamma_{1h}^* | \beta, \theta, \mathbf{z} = h, X, \mathbf{y}) = p(\gamma_{0h}^*, \gamma_{1h}^* | \mathbf{t}, \mathbf{y}, \mathbf{z} = h)$ and prior assumption $\gamma_{0h}^* \sim Beta(a_0, b_0)$ and $\gamma_{1h}^* \sim Beta(a_1, b_1)$. Therefore, for cluster $h$, we have

$$
p(\gamma_{0h}^{* (t)}, \gamma_{1h}^{* (t)} | \mathbf{t}^{(t-1)}, \mathbf{y}, \mathbf{z}^{(t-1)}) \propto p(\mathbf{y} | \mathbf{t}^{(t-1)}, \gamma_{0h}^{* (t-1)}, \gamma_{1h}^{* (t-1)}, \mathbf{z}^{(t-1)}) p(\gamma_{0h}^{* (t-1)}, \gamma_{1h}^{* (t-1)})
$$

$$
= \prod_{i, z_i = h} p(y_i | t_i^{(t-1)}, \gamma_{0h}^{* (t-1)}, \gamma_{1h}^{* (t-1)}) p(\gamma_{0h}^{* (t-1)}) p(\gamma_{1h}^{* (t-1)})
$$

$$
= (\gamma_{0h}^{* (t-1)}) n_{0ih}^{(t-1)} (1 - \gamma_{0h}^{* (t-1)}) n_{0ih}^{(t-1)} (\gamma_{1h}^{* (t-1)}) n_{0ih}^{(t-1)} (1 - \gamma_{1h}^{* (t-1)}) n_{0ih}^{(t-1)}
$$

$$
\times (\gamma_{0h}^{* (t-1)}) a_0^{-1} (1 - \gamma_{0h}^{* (t-1)}) b_0^{-1} (\gamma_{1h}^{* (t-1)}) a_1^{-1} (1 - \gamma_{1h}^{* (t-1)}) b_1^{-1}
$$

$$
= (\gamma_{0h}^{* (t-1)}) a_0 + n_{0ih}^{(t-1)} - 1 (1 - \gamma_{0h}^{* (t-1)}) b_0 + n_{0ih}^{(t-1)} - 1
$$

$$
\times (\gamma_{1h}^{* (t-1)}) a_1 + n_{0ih}^{(t-1)} - 1 (1 - \gamma_{1h}^{* (t-1)}) b_1 + n_{0ih}^{(t-1)} - 1
$$
where \( n_{01h}^{(t-1)}, n_{00h}^{(t-1)}, n_{10h}^{(t-1)}, n_{11h}^{(t-1)} \) are the number of counts of \((t_i^{(t-1)} = 0, y_i = 1), (t_i^{(t-1)} = 0, y_i = 0), (t_i^{(t-1)} = 1, y_i = 0), (t_i^{(t-1)} = 1, y_i = 1)\) among observations that satisfy \( z_i^{(t-1)} = h \).

The first line of the equation should contain \( p(t_i^{(t-1)}) \) and \( p(z_i^{(t-1)}) \) but they do not depend on misclassification probabilities and thus can be neglected up to proportionality. Consequently, the posterior distribution of misclassification probabilities for cluster \( h \) can be written as

\[
\begin{align*}
\gamma_{0h}^{(t)} | t^{(t-1)}, y &\sim Beta \left( a_0 + n_{01h}^{(t-1)}, b_0 + n_{00h}^{(t-1)} \right), \\
\gamma_{1h}^{(t)} | t^{(t-1)}, y &\sim Beta \left( a_1 + n_{10h}^{(t-1)}, b_1 + n_{11h}^{(t-1)} \right).
\end{align*}
\]

(3.7)

Note that equation (3.7) shows how to draw a misclassification rate from its posterior distribution for a cluster in each iteration. If a cluster is empty in iteration \( t \), it is not associated with any observation, thus does not play a part in the likelihood of observed data. There is therefore no need to draw from the posterior distribution of the misclassification probabilities of this cluster.

### 3.4.2 Posterior of clusters allocation parameters \( \pi \) and \( z \)

After updating cluster-specific parameters, we can update parameters that determine cluster allocation. We assume \( X \) follow a Dirichlet process mixture of Gaussians where latent cluster (Gaussian component) membership is represented by \( z \). As discussed in Section 3.2, this is equivalent to assuming \( z \) following discrete distribution \( \text{Discrete}(\pi) \) with stick breaking prior where each \( \pi_s \) is obtained though \( v_1, v_2, \ldots \) with

\[
\pi_s = v_s \prod_{l=1}^{s-1} (1 - v_l), \quad \text{where} \quad v_i \overset{iid}{\sim} Beta(1, \alpha).
\]

Using the method of Ishwaran and James (2001) discussed in Section 2.9.1, an infinite DP can be approximated with a truncated DP where the maximum number of cluster is \( H \). In
posterior updates, $\pi_s^{(t)}$ can still be obtained by $\pi_s^{(t)} = \nu_s^{(t)} \prod_{l=1}^{t-1} (1 - \nu_l^{(t)})$ but $\nu_s^{(t)}$ is sampled from its posterior

$$v_s^{(t)} \sim \text{Beta} \left(n_s^{(t-1)} + 1, \alpha^{(t-1)} + \sum_{j=s+1}^{k} n_j^{(t-1)} \right) \tag{3.8}$$

for $s = 1, \ldots, k - 1$. In each round, after updating the values for $\nu_1^{(t)}, \ldots, \nu_{k-1}^{(t)}$, we reset $\nu_k^{(t)} = 1$ to ensure $\pi_l^{(t)} = 0$ for all $l \geq k + 1$.

Note that (3.8) also depends on the previous round value of concentration parameter $\alpha^{(t-1)}$. Since we want to let the data inform the value of $\alpha$ and assumed it to follows Gamma hyperprior where $\alpha \sim \text{Gamma}(a_{\alpha}, b_{\alpha})$, $\alpha$ has posterior following Equation 2.9. As described in Section 2.9.2, to help improve mixing, when the truncated level is sufficiently large (which is the case of our experiment in Section 4), we use the update based on the marginal sampler instead. With marginal sampler, concentration parameter on iteration $t$, $\alpha^{(t)}$, follows

$$\eta^{(t-1)} \sim \text{Beta}(\alpha^{(t-1)} + 1, n), \quad u^{(t)} \sim \text{U}(0, 1),$$

$$b^{(t)} = b_{\alpha} - \log(\eta^{(t-1)}), \quad I^{(t)} = \mathbb{I} \left[ \frac{u^{(t)}}{1 - u^{(t)}} \geq \frac{a_{\alpha} + k - 1}{nb^{(t)}} \right],$$

$$\alpha^{(t)} \sim \text{Gamma}(a_{\alpha} + k - I^{(t)}, b^{(t)}) \tag{3.9}$$

Once $\pi_i^{(t)}$ is updated, $z_i^{(t)}$ can be updated from its posterior using the most recent values of $\pi_i^{(t)}$. The posterior sampling of $z_i$ is based on decomposition

$$p(z_i = h|x_i, y_i, t_i, \theta^*, \pi) \propto p(z_i = h, x_i, y_i | t_i, \theta^*, \pi).$$

$p(z_i = h, x_i, y_i | t_i, \theta^*, \pi)$ can further be decomposed as $p(x_i, y_i | t_i, \theta^*, \pi, z_i = h)p(z_i = h | \pi)$ where $p(z_i = h | \pi)$ is just $\pi_h$ and $p(x_i, y_i | t_i, \theta^*, \pi, z_i = h) = p(x_i | z_i = h, \mu^*, \Sigma^*)p(y_i | z_i = h, t_i, x_i, \gamma^*_{h1}, \gamma^*_{h0})$. Therefore, using $p_{ih}^{(t)}$, to denote the probability of the $i$th observation getting
assigned to cluster $h$ on iteration $t$, we can write

$$p_{ih}^{(t)} \propto f(x_i|\theta^{(t)}, z_i = h) f \left( y_i | z_i^{(t-1)} = h, t_i^{(t-1)} \right) \pi_{ih}^{(t)}.$$ \hspace{1cm} (3.10)

In other words, $z_i^{(t)} \sim \text{Discrete} \left( p_{i1}^{(t)}, ..., p_{iK}^{(t)} \right)$ where $p_{ih}^{(t)}$ is scaled such that $\sum_{h=1}^{K} p_{ih}^{(t)} = 1$ and the discrete distribution here is a multinomial. The factor \( f \left( y_i | z_i^{(t-1)} = h, t_i^{(t-1)} \right) \) looks long but is in fact just $\gamma_{0h}^{*}, \gamma_{1h}^{*}$, or $1 - \gamma_{1h}^{*}$ depending on the value of $y_i, t_i^{(t-1)}$ and $z_i^{(t-1)}$. This can be seen from the expansion

$$f \left( y_i | t_i^{(t-1)}, z_i^{(t)} = h, \gamma_{0h}, \gamma_{1h}, x_i \right) = \left( \gamma_{0h}^{*} \right)^{y_i \left( 1 - t_i^{(t-1)} \right)} \left( 1 - \gamma_{0h}^{*} \right)^{\left( 1 - y_i \right) \left( 1 - t_i^{(t-1)} \right)} \times \left( \gamma_{1h}^{*} \right)^{\left( 1 - y_i \right) \left( t_i^{(t-1)} \right)} \left( 1 - \gamma_{1h}^{*} \right)^{y_i \left( t_i^{(t-1)} \right)}.$$

3.4.3 Posterior of model parameters for responses $\beta$ and $t$

For the posterior inference of $\beta$, we have $p(\beta|\cdot) = p(\beta|X, t)$. For sampling convenience, we assume the covariates are standardised, which are common practice for Dirichlet process mixture models as discussed in (Gelman et al., 2013), and the coefficient for each covariate follows a normal prior where $\pi(\beta) = MVN(M_0, Q_0)$. We also assume that $t_i$ can be modeled by $x_i$ and $\beta$ via a generalised linear model with probit link where $g^{-1} = \Phi(\cdot)$ is the cumulative density function of a normal distribution. The posterior sampling of $\beta$ and $t_i$ can be made.
easier through introducing auxiliary variable $t_i^*$ which leads to new hierarchical model

$$y_i \sim \begin{cases} 
\text{Bern}(1 - \gamma_{ih}^*) & \text{if } t_i = 1 \\
\text{Bern}(\gamma_{0h}^*) & \text{if } t_i = 0.
\end{cases}$$

$$t_i = \begin{cases} 
1 & \text{if } t_i^* > 0 \\
0 & \text{if } t_i^* \leq 0,
\end{cases}$$

$$t_i^* = x_i \beta + \epsilon_i,$$

$$\epsilon_i \sim N(0, 1),$$

given $z_i = h$. Under the new hierarchical model, $t_i$ is deterministic conditional on the sign of auxiliary variable $t_i^*$ which implies that $p(\beta|t_i^*, t_i, x_i) = p(\beta|t_i^*, x_i)$. The posterior of $\beta$ can therefore be written as

$$p(\beta|t^*, t, X) = p(\beta|t^*, X) \propto p(t^*|\beta, X) \pi(\beta) = \pi(\beta) \prod_{i=1}^n N(t_i^*|x_i \beta, 1)$$

where $N(t_i^*|x_i \beta, 1)$ denotes that $t_i^*$ follows a normal distribution with mean $x_i \beta$ and variance 1. With a normal prior on $\beta$ where $\pi(\beta) \sim MVN(M_0, Q_0)$, we get posterior

$$\beta^{(t)}|t^{*(t-1)}, X \sim MVN(M^{(t)}, V^{(t)})$$

where

$$M^{(t)} = V^{(t)} \left( Q_0^{-1}M_0 + X^T t^{*(t-1)} \right), \quad V^{(t)} = \left( Q_0^{-1} + X^T X \right)^{-1}. \quad (3.11)$$

Since $\beta^{(t)}$ relies on $t^{*(t-1)}$, we need to update $t^*$ in each iteration of the MCMC sampler. For observation $i$, if we only condition $t_i^{*(t)}$ on $\beta^{(t)}$ and $x_i$, $t_i^{*(t)}$ would follow $N \left( x_i \beta^{(t)}, 1 \right)$. But
since the sign of $t^*_i$ determines $t_i$, $t^*_i$ cannot take any arbitrary value but must possess the right sign. In other words, $t^{* (t-1)}_i$ also depends on $t^{(t-1)}_i$ and has full conditional following a truncated normal where

$$t^{* (t)}_i | \beta^{(t)}, t^{(t-1)}_i, x_i \sim \begin{cases} TN \left( x_i \beta^{(t)}, 1 \right) & \text{if } t^{(t-1)}_i = 1, \\ TN \left( x_i \beta^{(t)}, 1 \right) & \text{if } t^{(t-1)}_i = 0. \end{cases}$$

Regarding $t^{(t)}_i$, it is conditionally dependent on $y_i, \beta^{(t)}, \gamma^{(t)}_0, \gamma^{(t)}_1, z^{(t)}_i$ and follows

$$p(t^{(t)}_i = 1 | y_i = 1, \beta^{(t)}, \gamma^{(t)}_0, \gamma^{(t)}_1, z^{(t)}_i, x_i) \propto p(y_i = 1 | t^{(t)}_i = 1, \gamma^{* (t)}_1)p(t^{(t)}_i = 1 | \beta^{(t)}, x_i)$$

and

$$p(t^{(t)}_i = 1 | y_i = 0, \beta^{(t)}, \gamma^{(t)}_0, \gamma^{(t)}_1, z^{(t)}_i, x_i) \propto p(y_i = 0 | t^{(t)}_i = 1, \gamma^{* (t)}_1)p(t^{(t)}_i = 1 | \beta^{(t)}, x_i)$$

where $p(t^{(t)}_i = 1 | \beta^{(t)}, x_i) = g^{-1}(x_i \beta^{(t)})$ and $p(y_i = l | t^{(t)}_i = s, \gamma^{(t)}_1)$ with $l, s \in \{0, 1\}$ representing the probability of correct classification or misclassification for the $i$th observation. Therefore

$$t^{(t)}_i \sim \begin{cases} \text{Bern} \left( \frac{1 - \gamma^{* (t)}_1}{(1 - \gamma^{* (t)}_1) \Phi(x_i \beta^{(t)}) + \gamma^{* (t)}_1[1 - \Phi(x_i \beta^{(t))}]} \right) & \text{if } y_i = 1, \\ \text{Bern} \left( \frac{\gamma^{* (t)}_1 \Phi(x_i \beta^{(t)})}{\gamma^{* (t)}_1 \Phi(x_i \beta^{(t)}) + [1 - \gamma^{* (t)}_1][1 - \Phi(x_i \beta^{(t)})]} \right) & \text{if } y_i = 0. \end{cases}$$

Equation (3.12) intuitively makes sense because when $\gamma^{* (t)}_1 = \gamma^{* (t)}_0 = 0$, the probability of the Bernoulli distribution becomes 1 when $y_i = 1$ and becomes 0 when $y_i = 0$, which corresponds to the case of no misclassification. If we also only use $X$ for the clustering, (3.12) would become $t^{(t)}_i \sim \text{Bern} \left( \Phi(x_i \beta^{(t)} \right)$, which is just a regular generalised linear model.

In summary, the updates regarding $\beta$ and $t$ in DP-PMM take the steps of
1. sample $\beta(t)|t^{(t-1)}, X \sim MVN(M(t), V(t))$ where $M(t) = V(t) (Q_0^{-1} M_0 + X^T t^{(t-1)})$ and $V(t) = (Q_0^{-1} + X^T X)^{-1}$.

2. sample $t^*_i(t)|\beta(t), t_i^{(t-1)}, x_i \sim \begin{cases} TN(x_i \beta(t), 1) \in (0, \infty), & \text{if } t_i^{(t-1)} = 1; \\ TN(x_i \beta(t), 1) \in (-\infty, 0], & \text{if } t_i^{(t-1)} = 0. \end{cases}$

3. sample $t_i^{(t)}|t^*_i(t), \beta(t), z_i \sim \begin{cases} Bern \left( \frac{(1-\gamma^*_i(t)) \phi(x_i \beta(t))}{(1-\gamma^*_i(t)) \phi(x_i \beta(t)) + \gamma^*_i(t) |1-\phi(x_i \beta(t))|} \right), & \text{if } y_i = 1; \\ Bern \left( \frac{\gamma^*_i(t) \phi(x_i \beta(t))}{\gamma^*_i(t) \phi(x_i \beta(t)) + (1-\gamma^*_i(t)) |1-\phi(x_i \beta(t))|} \right), & \text{if } y_i = 0. \end{cases}$

**Algorithm for DP-PMM**

We summarise all steps of DP-PMM in Algorithm 8. Note that label switching might occur during the sampling process and can be dealt with post-sampling techniques which we discuss.
Deal with potential label switching of clusters and adjust the values of 
\( \mu^*, \Sigma^*, \gamma_0^*, \gamma_1^* \) accordingly;
3.5 Predictive inference

Using $D$ to denote data where $D = (x_i, y_i)_{i=1}^n$, the posterior can be expressed through a conditional expectation,

$$
\mathbb{E}(t_{n+1}|x_{n+1}, D) = \mathbb{E}[\mathbb{E}(t_{n+1}|x_{n+1}, \theta_{1:n})|D]
$$

where $\theta_{1:n}$ denote the parameters that generates the observed data. This can be approximated by Monte carlo average using $R$ posterior samples of $\theta_{1:n}$ which in the interest of $t_{n+1}$ are the probit regression for true responses, therefore

$$
\mathbb{E}(t_{n+1}|x_{n+1}, D) \approx \frac{1}{R} \sum_{r=1}^R \mathbb{E}(t_{n+1}|x_{n+1}, \theta_{1:n}^{(r)}) = \frac{1}{R} \sum_{r=1}^R \Phi \left( x_{n+1} \beta^{(r)} \right).
$$

Regarding the predictive expectation of the misclassification probabilities, we would estimate $E[\gamma_{1,n+1}|D, x_{n+1}]$ if $\hat{t}_{n+1}|x_{n+1}, D = 1$ and $E[\gamma_{0,n+1}|D|x_{n+1}]$ if $\hat{t}_{n+1}|x_{n+1}, D = 0$. Similarly, we can write

$$
E[\gamma_{s,n+1}|D, x_{n+1}] = \mathbb{E}[\mathbb{E}(\gamma_{s,n+1}|x_{n+1}, \theta_{1:n})|D] \approx \frac{1}{R} \sum_{r=1}^R \mathbb{E} \left( \gamma_{s,n+1}|x_{n+1}, \theta_{1:n}^{(r)} \right) = \frac{1}{R} \sum_{r=1}^R \sum_{h=1}^K \sum_{s=1}^{s_{sh}} p_{n+1,h}^{(r)} \gamma_{sh}^{(r)}
$$

where $s = 1$ or $s = 0$ depends on predicted value of $t_{n+1}$ and

$$
p_{n+1,h}^{(r)} \propto f(x_{n+1} | \theta^{(r)}), z_{n+1} = h) \pi_h^{(r)}.
$$
Chapter 4

Experiment on real data

To test how DP-PMM performs, we apply it to a real dataset. Ideally we apply DP-PMM to a dataset that have covariates and contaminated observed responses for modelling and true responses for comparison. However, real-world datasets with both true and observed responses are hard to obtain. Therefore, in our experiments we use a real dataset where the responses at hand are treated as true responses and observed responses are obtained by injecting artificial misclassification on the true responses. The dataset we use is the phoneme dataset - a dataset extracted from digitised speech from the TIMIT database and is used as a resource for speech recognition (https://sci2s.ugr.es/keel/description.php). The dataset contains 5405 observations and five covariates and all covariates are continuous attributes characterising each vowel. The covariates are the amplitudes of the five first harmonics $AHi$, normalised by the total energy $Ene$: $aa$, $ao$, $dcl$, $iy$, $sh$. The binary responses at hand correspond to the classification of the class of the vowel with 0 for nasal vowel and 1 for oral vowels. In the phoneme data at hand, the majority of the observations (about 70%) are nasal vowels.

We take the responses in the phoneme data at hand as true responses $t$ and inject artificial contamination under a range of misclassification probabilities to obtain observed responses.
Our proposed DP-PMM for misclassified binary data is then applied to \((X, y)\). To model the likelihood of responses, we assume a probit model regressing true responses \(t\) on \(X_1, X_2, X_3, X_4, X_5\). The predictors used are \(X_1, X_2, X_3, X_4, X_5\) where the \(X_1\) is the amplitude of harmonics \(aa\) normalised by total energy and \(X_2, X_3, X_4, X_5\) are the same measurement for \(ao, dcl, iy, sh\). For comparison, we use the estimated regression coefficients obtained from the probit regression regressing true responses \(t\) on \(X_1, X_2, X_3, X_4, X_5\) as the baseline of true values.

Before fitting DP-PMM, the phoneme dataset is first split into training set containing 80% of the data and test set containing the remaining 20% of data. The results for estimated regression coefficients are obtained from \((x_i, y_i)\)'s in the training set and the results that reflect predictive power are obtained from comparing model predicted responses and true responses in the test set.

### 4.1 Set up

In this Section, we discuss how we set up our experiments and inject artificial misclassification in the phoneme data mentioned in Section 9. There are 5404 observations in total and 4323 observations in the training data. A visualisation of the distribution of covariates and responses of the data is shown in 4.1. As we can see in the plot, about 30% of the responses are 1. None of the covariates appear to follow any single distribution but all seem plausible to be approximated by a mixture of normal distributions.
Figure 4.1: Histogram of each covariates and responses at hand $t$.

We also plot the covariates pairwise with colour-coded true responses in Figure 4.2. From Figure 4.2, we see that the phoneme dataset seem to have interesting complex structure.
We center the covariates to simplify prior specification for the Gaussian components. To obtain the observed responses $y_i$ for each observation $i$, we flip its true response $t_i$ with probability $\gamma_{t_i,i}$ where $\gamma_{t_i,i}$ is a function of the covariates. The underlying assumption is that the misclassification probabilities can be modelled by the covariates. The function used for $\gamma_{t_i,i}$ is different for each misclassification setting and we discuss them in Section 4.1.1. DP-PMM is applied to the resulting observed data $(X, y)$. 

Figure 4.2: Pairwise plot of covariates for phoneme dataset with colour-coded true responses (pink for 0 and green for 1).
4.1.1 Injection of misclassification

We conduct three experiments with low, moderate, and high level of misclassification, along with applying DP-PMM to the data at hand with no misclassification. The misclassification probabilities are a function of the covariates in each case and varies across the observations. For each misclassification setting, the functions that generate misclassification are different and so are the priors used for DP-PMM. In the first setting of our experiments, we simulate a low misclassification setting where covariates-specific misclassification are injected in the phoneme dataset through flipping true response \( t_i \) with probability \( \gamma_{t_i,i} \) where

\[
\gamma_{t_i,i} = \frac{1}{1 + e^{-(x_{i5} - 4)}}.
\]

The resulting average level of misclassification is \( \bar{\gamma}_0 \approx 0.021, \bar{\gamma}_1 \approx 0.030 \).

In the case where we inject moderate level of misclassification, we flip true response \( t_i \) with probability \( \gamma_{t_i,i} \) where

\[
\gamma_{t_i,i} = \frac{1}{1 + e^{-(0.8x_{i4} - 3)}}.
\]

The resulting average level of misclassification is \( \bar{\gamma}_0 \approx 0.066, \bar{\gamma}_1 \approx 0.091 \).

In the last case of high misclassification, covariates-specific misclassification are injected through flipping true response \( t_i \) with probability \( \gamma_{t_i,i} \) where

\[
\gamma_{t_i,i} = \frac{1}{1 + e^{-(0.5x_{i4} - 2)}}.
\]

The resulting average level of misclassification is \( \bar{\gamma}_0 \approx 0.132, \bar{\gamma}_1 \approx 0.169 \).
4.1.2 Priors used

In DP-PMM, we assume the maximum number of clusters is $K = 70$. From the posterior analysis of the number of occupied clusters in different iterations, we confirm this maximum bound is sufficient since in nearly all cases the number of occupied clusters is less than 35. For each cluster, it is assumed to a multivariate Gaussian with covariance $\Sigma_h$ and mean $\mu_h$ where

$$
\Sigma_h^* \sim W^{-1}(\mu_h = 0.5I_p, \Psi_h = 1.5I_p),
$$

$$
\mu_h|\Sigma_h^* \sim MVN(0, \Sigma_h^*/\Psi_h),
$$

with $I_p$ denoting the $p \times p$ identity matrix ($p = 5$ here as there are 5 covariates).

For concentration parameters $\alpha$, we assume that it follows a Gamma distribution where

$$
\alpha \sim Gamma(0.01, 0.01).
$$

The hyper-parameters are chosen such that the mean and variance of $\alpha$ is 1 and 100. We assume a hyperprior for $\alpha$ because $\alpha$ influences the number of clusters and we want to let the data to inform about the appropriate value of $\alpha$.

For the priors for misclassification probabilities, we assume $\gamma_{0h}^*$ and $\gamma_{1h}^*$ each follow a Beta prior. When there is no misclassification in the data, we assume $\gamma_{0h}^* \sim Beta(1, 1000), \gamma_{1h}^* \sim Beta(1, 1000)$. In other cases, we choose a Beta prior such that the prior broadly covers the histogram of the true $\gamma_{0i}, \gamma_{1i}$ for all observation $i$ in the training data. This way of setting prior roughly ensures fairness as some methods we compare DP-PMM to assume known misclassification probabilities $\gamma_0, \gamma_1$ and we use the mean true misclassification probabilities $\bar{\gamma}_0, \bar{\gamma}_1$ for the known $\gamma_0, \gamma_1$. 

128
4.1.3 Gibbs sampling

We run our Gibbs sampling for 5000 iterations with 3000 burn-ins. To check that our algorithm has converged, we check the traceplots of the parameters. In Figure 4.3, we show the traceplots for concentration parameter $\alpha$ and $\beta_0$. The traceplots are similar for the other parameters and we omit for brevity.

![Traceplot of $\alpha$](image1.png) ![Traceplot of $\beta_0$](image2.png)

Figure 4.3: Traceplots for concentration parameter $\alpha$ (left) and $\beta_0$ (right)

4.1.4 Consideration on label switching

The label switching problem refers to the fact that under symmetric prior (same prior for each cluster), the resulting posterior will be invariant when labels are permuted. Since in our case the posterior is proportional to the product of a symmetric likelihood with a symmetric prior, the marginal distribution will be all the same. The components are exchangeable as all the marginals on $\theta_j$ are identical. Though the estimated $\beta$ will not differ under the presence of label switching, other cluster-specific parameters such as inferred misclassification probabilities, Gaussian means and covariance matrices would be affected.

We deal with potential label switching in the post-sampling phase by reordering the MCMC samples using an equivalence class representative algorithm.
The big idea behind *equivalence class representation (ECR)* algorithm is to find the optimal permutation to undo label switching by pivoting on the cluster assignment, i.e., a pivot allocation vector (Papastamoulis and Iliopoulos, 2010). The optimal permutation on simulated MCMC sample $t$ can be defined as the one that minimises the distance of cluster assignment under it compared to the ‘true allocation’. Let $g = (g_1, g_2, ..., g_n)^T$ denote the true allocation, we can write the objective as maximising the number of observations matched after permutation, denoted by \( \sum_{i=1}^{n} 1[g_i = p_t(z_i^{(t)})] \). Since the ‘true allocation’ $g$ is not known, it needs to be estimated. The ECR algorithm is a method proposed by Papastamouilis and Iliopoulos (2010) that maximises total number of observations matched after permutation by finding a pivot iteratively. In each iteration, the mode cluster assignment from previous iteration is used as the new pivot until no improvement can be made. We apply the iterative ECR to DP-PMM using *R* Package `label.switching`. More discussion on label switching and the ECR algorithm can be found in Appendix 4.1.4.

### 4.2 Evaluation criteria

To evaluate DP-PMM, we use various criteria to assess the inferential and predictive ability of DP-PMM. We use the training set of phoneme data to obtain inferential results and the test set to obtain predictive results.

#### 4.2.1 Criteria for inferential ability

Since the main interest of inference is the true coefficients that parametrise the probit regression model for true responses, primarily we use the discrepancy between the estimated $\beta$ and its true values as the criteria for inferential ability. We use the regression coefficient of a probit regression model fitted to the uncontaminated training data as proxy of the true values. We compare the estimated values of $\beta = (\beta_0, \beta_1, \beta_2)^T$ to their true values, as well
as the estimated values of $\beta$ obtained by naively fitting the probit regression to observed responses. Since DP-PMM gives an estimate of $\beta_0, \beta_1, \beta_2$ in each MCMC iteration, we can obtain a credible interval for each parameter of the probit model.

Beside the regression coefficients, we also check the misclassification probabilities learned by DP-PMM. Since DP-PMM outputs a set of cluster assignments and the misclassification probabilities of each cluster in each iteration of the MCMC, we can find the an estimated misclassification probability of each observation in each iteration. This is invariant to label switching and provide us a posterior distribution of misclassification probability of each observation. This information of the misclassification probability of each observation could be of interest to practitioners and compared with the true misclassification probabilities to evaluate the effectiveness of DP-PMM.

### 4.2.2 Criteria for predictive ability

We are interested in investigating how well each method estimates the true regression coefficients and predicts out-of-sample observations. To evaluate predictive power, we use the typical criterion of classification accuracy (or balanced accuracy), in which the proportion of correctly classified observations is calculated among all observations in test data. In addition to balanced accuracy, we also want to evaluate the accuracy within each class, therefore also adopt the AUC and F-measure. The relevant statistics that is used to calculate F-measure and AUC are sample true positive rate (TPR) and sample false positive rate (FPR) as described in Section 1.6.3.

Another metric that we also include in the comparison is the average log loss of predictive probability of the test data. This is defined as

$$-rac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} [t_i \log(p_i) + (1 - t_i) \log(1 - p_i)]$$
where \( p_i \) is the predicted probability that the true response is equal to one for test data (Murphy, 2012). The lower the log loss is, the higher the predictive power of the model.

### 4.3 Results and discussions

From DP-PMM, we naturally obtain posterior distributions of estimated regression coefficients of the probit regression model. In addition, we also have many parameters such as the misclassification probabilities for each cluster and the component-specific parameters that generate \( X \) in each cluster. The misclassification probabilities for each cluster can be used to check the reasonableness model output. The component-specific parameters provide a mean to generate \( X \). Furthermore, we can also check the number of occupied clusters among the MCMC runs to confirm if the upper limit for the number of components we set, \( H \), is large enough.

#### 4.3.1 Case 1: no misclassification

Before we inject misclassification to the data and apply DP-PMM, we check that DP-PMM gives sensible inference when there is no injected misclassification. We apply DP-PMM to the entire phoneme data at hand assuming a normal prior for \( \beta \) centered around 0 and with standard deviation of 5 for all coefficients. Since there is no misclassification in the data in this case, we assume that \( \gamma_{0h}, \gamma_{1h} \) follows \( \text{Beta}(1,1000) \) for all clusters where \( h \in \{1, 2, \ldots, H\} \).

Directly applying a probit regression model to the training data of phoneme dataset, we obtain \( \hat{\beta} = (-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T \). As shown in Figure 4.4, these ‘true values’ lie in the 95% credible interval given by DP-PMM.
Figure 4.4: 95% credible interval (black line) for estimated $\beta_j$ from DP-PMM in the phoneme dataset when there is no misclassification injected compared to ‘true values’ from probit regression $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ indicated by red line.

In this case, there is no misclassification in training data of phoneme dataset and from Figure 4.5, we can see that the average $\gamma_{0h}, \gamma_{1h}$ for all clusters from DP-PMM are all very small and close to 0.

Figure 4.5: Histogram of average $\gamma_{0h}$ in nonempty clusters (left) and histogram of average $\gamma_{1h}$ in all clusters (right) from DP-PMM in the phoneme dataset when there is no misclassification injected in training data.

Note that for this dataset, we assumed there are 70 components at max and not all components need to be occupied. In fact, if we check the number of clusters (nonempty
components) in all iterations as shown in the left plot of Figure 4.6, we see that the number of clusters are always less than 35. This confirms that 70 is an appropriate component number to assume for this dataset. The right plot of Figure 4.6 shows the trace plot of the number of clusters, which suggests convergence. The trace plots of other parameters shows similar pattern that also suggest convergence. Regarding clustering results, Figure 4.7 shows the heat map of posterior similarity matrix of clustering on phoneme training data.

Figure 4.6: Histogram of number of nonempty clusters among 5000 MCMC runs when there is no misclassification injected in the training data of phoneme dataset.

Figure 4.7: Heat map of posterior similarity matrix of clustering on phoneme training data when there is no misclassification injected.
4.3.2 Case 2: low misclassification

Now we inject misclassification at different levels to the phoneme training data and apply DP-PMM to contaminated training data. We compare DP-PMM to the EM method, the post-fit adjustment of regression coefficients, MC-SIMEX, and maximum estimated likelihood discussed in Section 1.3. For fairness, we do not compare our method to the methods that assume validation data. We also include inference and prediction results from directly fitting a probit regression to the observed data and results fitting a probit regression to true responses.

Recall that in the case of low level of misclassification, we flip the true response of observation $i$ with probability $\gamma_{t,i}$ where $\gamma_{t,i} = \frac{1}{1+e^{-(x_i^5-4)}}$. This leads to $\bar{\gamma}_0 \approx 0.022, \bar{\gamma}_1 \approx 0.030$ as the average level of misclassification in the training data. In Figure 4.8, we plot the misclassification probabilities as a function of $X_5$ and in Figure 4.9, we visualise the histogram of true $\gamma_{0i}, \gamma_{1i}$ for all observation $i$ in training data.

![Misclassification probability as a function of $x_5$ in training data](image)

Figure 4.8: Misclassification probabilities as a function of $X_5$ in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.
Figure 4.9: Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{0i}$ (right) for all observation $i$ in phoneme training data when there is low misclassification. Blue line indicates the Beta prior used for $\gamma_{0h}^*, \gamma_{1h}^*$ in DP-PMM.

The blue line in Figure 4.9 is the Beta prior we used for $\gamma_{0h}, \gamma_{1h}$ for DP-PMM. In this setting of low misclassification, we choose $\gamma_{0h}^* \sim Beta(0.6, 27), \gamma_{1h}^* \sim Beta(0.5, 16)$ so that the priors roughly cover the histogram of true misclassification probabilities. The results for estimated $\beta_j$'s from fitting DP-PMM to observed data are shown in Figure 4.4. From the plot we can see that the 95% credible interval estimate of $\beta_j$'s given by DP-PMM contain the ‘true values’ of probit regression coefficients, which are indicated by the red lines.
Figure 4.10: 95% credible interval for estimated $\beta_j$ from DP-PMM fitted phoneme training data with low misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ from probit regression fitted to the uncontaminated training data indicated by the red lines.

Regarding the misclassification probabilities, DP-PMM outputs the estimates for $\gamma_{0h}, \gamma_{1h}$ for each cluster in each iteration. Using these results about misclassification probabilities observation allocation, we can estimate the distribution of $\gamma_{0h}, \gamma_{1h}$ for all iterations and obtain misclassification probability for each observation. In Figure 4.11 we plot the histogram of estimated average misclassification probabilities for all clusters from DP-PMM. As we can see, when there is low level of misclassification injected to the training data of phoneme dataset, DP-PMM picks up small misclassification probabilities that peak near the actual $\gamma_{0i}, \gamma_{1i}$ shown in Figure 4.9.
Figure 4.11: Histogram of estimated average misclassification probabilities for all clusters from DPMM in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.

From DP-PMM, since we have misclassification probabilities for all clusters in each iteration and cluster assignments for all observations in each iteration, we can estimate the misclassification probability for each observation in each iteration. In Figure 4.12, we plot the average misclassification probabilities of all observations against different values of $X_5$ in yellow dots. We use local polynomial regression fitting (fitting at a point uses neighbour points weighted by distance) to approximate misclassification probability as a function $X_5$ in blue line and compare it to the actual misclassification probabilities in red line. Though in some parts, we see high misclassification probabilities with higher value of $X_5$, we do not observe blue line or yellow dots in the shape of red line. Besides that these are point estimates and there is additional uncertainty in the misclassification probabilities, this is also likely because of uneven distribution of different values of $X_5$ shown in Figure 4.1 where most observations have values near 0.
Figure 4.12: Estimated mean misclassification probabilities as a function $X_5$ in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.

For clustering results, we visualise the heatmap of posterior similarity matrix in Figure 4.13.

Figure 4.13: Heatmap of posterior similarity matrix for clustering results from DPMM in case 2 of experiments when there is low level of misclassification injected to the training data of phoneme dataset.

In terms of inference, if a probit regression is fitted to the uncontaminated training data,
the estimated regression coefficients is \( \hat{\beta} = (-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T \). If a probit regression is fitted to the observed training data with low level of misclassification, the estimated regression coefficients is \( \hat{\beta} = (-0.545, -0.312, -0.229, 0.336, 0.424, 0.290)^T \).

For DP-PMM, we have a posterior distribution of regression coefficients and report the posterior mean in Table 1.3. In the same table, we further summarise the estimated regression coefficients from the EM method, the post-fit adjustment of regression coefficients, MC-SIMEX, and maximum estimated likelihood, all assuming a probit regression model for the true responses. The results from fitting a probit regression to uncontaminated training data and observed data are also included as ‘GLM to T’ and ‘GLM to Y’. Applied to data with low level of misclassification, all methods seem to do fine in estimating the probit regression coefficients, with MEL relatively worse than others. The prediction results applied to uncontaminated test data are shown in Table 4.2 and all methods perform similarly in terms of accuracy, ROC, F-measure, and log loss of predictive probability.

<table>
<thead>
<tr>
<th></th>
<th>est. ( \beta_0 )</th>
<th>est. ( \beta_1 )</th>
<th>est. ( \beta_2 )</th>
<th>est. ( \beta_3 )</th>
<th>est. ( \beta_4 )</th>
<th>est. ( \beta_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>-0.609</td>
<td>-0.387</td>
<td>-0.247</td>
<td>0.400</td>
<td>0.465</td>
<td>0.312</td>
</tr>
<tr>
<td>EM</td>
<td>-0.607</td>
<td>-0.346</td>
<td>-0.219</td>
<td>0.369</td>
<td>0.463</td>
<td>0.318</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>-0.626</td>
<td>-0.356</td>
<td>-0.217</td>
<td>0.357</td>
<td>0.451</td>
<td>0.309</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>-0.609</td>
<td>-0.394</td>
<td>-0.225</td>
<td>0.372</td>
<td>0.456</td>
<td>0.312</td>
</tr>
<tr>
<td>MEL</td>
<td>-0.529</td>
<td>-0.298</td>
<td>-0.189</td>
<td>0.305</td>
<td>0.396</td>
<td>0.271</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>-0.545</td>
<td>-0.312</td>
<td>-0.229</td>
<td>0.336</td>
<td>0.424</td>
<td>0.290</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>-0.610</td>
<td>-0.335</td>
<td>-0.253</td>
<td>0.380</td>
<td>0.470</td>
<td>0.321</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of the average estimates of \( \beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5 \) in phoneme dataset under low misclassification setting where \((\gamma_0, \gamma_1) = (0.022, 0.030)\). The result from DP-PMM refers to the mean of estimated \( \beta \) over 5000 MCMC runs.
Table 4.2: Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under low misclassification setting where \((\bar{\gamma}_0, \bar{\gamma}_1) = (0.022,0.030)\). Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated \(\beta\) in every 200 run out of 5000 MCMC runs in total.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>ROC</th>
<th>F-measure</th>
<th>Log Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>0.823</td>
<td>0.756</td>
<td>0.507</td>
<td>0.458</td>
</tr>
<tr>
<td>EM</td>
<td>0.824</td>
<td>0.760</td>
<td>0.521</td>
<td>0.459</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>0.824</td>
<td>0.752</td>
<td>0.488</td>
<td>0.461</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>0.823</td>
<td>0.755</td>
<td>0.502</td>
<td>0.459</td>
</tr>
<tr>
<td>MEL</td>
<td>0.824</td>
<td>0.752</td>
<td>0.492</td>
<td>0.465</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>0.824</td>
<td>0.757</td>
<td>0.508</td>
<td>0.459</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>0.824</td>
<td>0.759</td>
<td>0.512</td>
<td>0.458</td>
</tr>
</tbody>
</table>

4.3.3 Case 3: moderate misclassification

In this section we show the same experiment outputs but when there is moderate level of misclassification in the training data of phoneme dataset. In this case, the true responses are flipped with probability \(\gamma_{t,i} = \frac{1}{1+e^{-(\eta t_i-\theta)}}\) and the overall level of misclassification is \(\bar{\gamma}_0 \approx 0.066, \bar{\gamma}_1 \approx 0.091\). We plot the histogram of estimated average misclassification probabilities for all clusters in Figure 4.14. The priors for misclassification probabilities in this case are \(\gamma_{0h} \sim Beta(0.9, 13), \gamma_{1h} \sim Beta(1, 10)\) and are indicated by the blue line in Figure 4.14. As we can see from the plot, the priors seem to cover the histogram of all misclassification probabilities.
Figure 4.14: Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{0i}$ (right) for all observation $i$ in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset. Blue line indicates the Beta prior used for $\gamma_{0h}, \gamma_{1h}$ in DP-PMM.

The results for estimated $\beta_j$’s from fitting DP-PMM to phoneme training data with moderate level of misclassification are shown in Figure 4.15. From the plot we can see that the 95% credible interval estimate of $\beta_j$’s given by DP-PMM contain the ‘true values’ of probit regression coefficients, which are indicated by the red lines.

Figure 4.15: 95% credible interval for estimated $\beta_j$ from DP-PMM fitted to phoneme training data with moderate level of misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ from probit regression fitted to the uncontaminated training data indicated by the red lines.
Regarding the learned misclassification probabilities, we plot the histogram of estimated average misclassification probabilities for all clusters from DP-PMM in Figure 4.16. From the plot, we can see a range of misclassification probabilities with peaks for $\gamma_{0h}$ centers near 0.05. This matches the actual peak of true $\gamma_{0i}$ shown in Figure 4.14. For $\gamma_{1h}$, DP-PMM picks up generally higher values for each cluster than the actual values for $\gamma_{1i}$’s. Using the misclassification probabilities for all clusters in each iteration and cluster assignments for all observations in each iteration, we can also estimate the misclassification probability for each observation in each iteration. Figure 4.17 plots the average misclassification probabilities of all observations against different values of $X_4$ in yellow dots and use local polynomial regression fitting to estimate a smooth line in blue. We compare this to the actual misclassification probabilities that is a function $X_4$ indicated in the red line. Like in the case of low level of misclassification, we see high misclassification probabilities with higher value of $X_4$, in some parts but we do not observe blue dots in the shape of red line. Besides that the uncertainty in point estimates, upon examining the distribution of $X_4$ in Figure 4.1, we also observe uneven distribution of values of where the majority of observations have values between -0.5 and 1.

Figure 4.16: Histogram of estimated average misclassification probabilities for all clusters from DP-PMM in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset.
The heatmap of posterior similarity matrix of clustering results in this case is shown in Figure 4.13.

In this experiment where there are moderate level of misclassification, a probit regression
naively fitted to the observed training data (with misclassified responses) yields estimated regression coefficients: \( \hat{\beta} = (\begin{array}{cccccc}
-0.497 & -0.300 & -0.331 & 0.308 & 0.398 & 0.238
\end{array})^T. \) The results for average estimates of \( \hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4, \hat{\beta}_5 \) are given in Table 4.3 and the results for prediction using test data are given in Table 4.4. All methods including DP-PMM gives better estimates of probit regression coefficients than GLM to T except MEL, which seems to give estimates farther away from the ‘true values’ estimated by fitting probit regression to contaminated training data. For prediction, all methods perform similarly in terms accuracy, but EM, the post-fit adjustment of regression coefficients, and DP-PMM perform better in terms of ROC, F-measure, and log loss of predictive probabilities.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \text{est. } \hat{\beta}_0 )</th>
<th>( \text{est. } \hat{\beta}_1 )</th>
<th>( \text{est. } \hat{\beta}_2 )</th>
<th>( \text{est. } \hat{\beta}_3 )</th>
<th>( \text{est. } \hat{\beta}_4 )</th>
<th>( \text{est. } \hat{\beta}_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>-0.609</td>
<td>-0.387</td>
<td>-0.247</td>
<td>0.400</td>
<td>0.465</td>
<td>0.312</td>
</tr>
<tr>
<td>EM</td>
<td>-0.627</td>
<td>-0.301</td>
<td>-0.240</td>
<td>0.366</td>
<td>0.498</td>
<td>0.401</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>-0.607</td>
<td>-0.326</td>
<td>-0.240</td>
<td>0.348</td>
<td>0.470</td>
<td>0.370</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>-0.714</td>
<td>-0.382</td>
<td>-0.258</td>
<td>0.387</td>
<td>0.482</td>
<td>0.367</td>
</tr>
<tr>
<td>MEL</td>
<td>-0.461</td>
<td>-0.198</td>
<td>-0.151</td>
<td>0.216</td>
<td>0.307</td>
<td>0.241</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>-0.497</td>
<td>-0.300</td>
<td>-0.331</td>
<td>0.308</td>
<td>0.398</td>
<td>0.238</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>-0.686</td>
<td>-0.353</td>
<td>-0.276</td>
<td>0.401</td>
<td>0.530</td>
<td>0.312</td>
</tr>
</tbody>
</table>

Table 4.3: Comparison of the average estimates of \( \hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4, \hat{\beta}_5 \) in phoneme dataset under moderate misclassification setting where \((\bar{\gamma}_0, \bar{\gamma}_1) = (0.066, 0.091)\). The result from DP-PMM refers to the mean of estimated \( \beta \) over 5000 MCMC runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>accuracy</th>
<th>ROC</th>
<th>F-measure</th>
<th>log loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>0.823</td>
<td>0.756</td>
<td>0.507</td>
<td>0.458</td>
</tr>
<tr>
<td>EM</td>
<td>0.823</td>
<td>0.758</td>
<td>0.512</td>
<td>0.459</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>0.824</td>
<td>0.757</td>
<td>0.498</td>
<td>0.459</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>0.824</td>
<td>0.746</td>
<td>0.450</td>
<td>0.462</td>
</tr>
<tr>
<td>MEL</td>
<td>0.824</td>
<td>0.746</td>
<td>0.443</td>
<td>0.484</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>0.823</td>
<td>0.752</td>
<td>0.480</td>
<td>0.464</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>0.822</td>
<td>0.753</td>
<td>0.493</td>
<td>0.459</td>
</tr>
</tbody>
</table>

Table 4.4: Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under moderate misclassification setting where \((\bar{\gamma}_0, \bar{\gamma}_1) = (0.066, 0.091)\). Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated \( \beta \) in every 200 run out of 5000 MCMC runs in total.
4.3.4 Case 4: high misclassification

In the setting with high level of misclassification, the misclassifications are generated by flipping true responses with probability $\gamma_{t,i} = \frac{1}{1+e^{-0.5x_i^{2.2}}}$, and the resulting overall level of misclassification is $\bar{\gamma}_0 \approx 0.132$, $\bar{\gamma}_1 \approx 0.169$. We plot the histogram of estimated average misclassification probabilities for all clusters in Figure 4.14. The priors for misclassification probabilities in this case are $\gamma_{0h}^* \sim Beta(0.9, 6)$, $\gamma_{1h}^* \sim Beta(0.8, 4)$ indicated by the blue line in Figure 4.19. As we can see from the plot, the priors seem to cover the histogram of all misclassification probabilities.

Figure 4.19: Histogram of true $\gamma_{0i}$ (left) and true $\gamma_{0i}$ (right) for all observation $i$ in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset. Blue line indicates the Beta prior used for $\gamma_{0h}, \gamma_{1h}$ in DP-PMM.

The results for estimated $\beta_j$’s from fitting DP-PMM to phoneme training data with high level of misclassification are shown in Figure 4.4. Like before, we can see that the 95% credible interval estimate of $\beta_j$’s given by DP-PMM contain the ‘true values’ of probit regression coefficients, which are indicated by the red lines.
Figure 4.20: 95% credible interval for estimated $\beta_j$ from DP-PMM fitted to phoneme training data with high level of misclassification compared to ‘true values’ $(-0.609, -0.387, -0.247, 0.400, 0.465, 0.312)^T$ from probit regression fitted to the uncontaminated training data indicated by the red lines.

Regarding the misclassification probabilities, we have the histogram of estimated average misclassification probabilities for all clusters from DP-PMM in Figure 4.11 and the average misclassification probabilities of all observations against different values of $X_4$ in Figure 4.22. Compared to the histogram of actual individual misclassification probabilities, the histogram of misclassification probabilities for different clusters covers similar range but have higher peaks. Looking at the estimated misclassification probabilities against $X_4$ in Figure 4.22, we also see high misclassification probabilities with higher value of $X_4$ in some parts like the underlying true misclassification function in red line. In overall, we do not observe blue dots in the shape of red line. As in previous case, we first note that these are point estimates and there is additional uncertainty in the misclassification probabilities. Another possible cause is the uneven distribution of different values of $X_4$ shown in Figure 4.1 where most observations have values not too far from 0.
Figure 4.21: Histogram of estimated average misclassification probabilities for all clusters from DPMM in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset.

Figure 4.22: Estimated mean misclassification probabilities as a function $X_4$ in case 4 of experiments when there is high level of misclassification injected to the training data of phoneme dataset.

The heatmap of posterior similarity matrix that visualise clustering results when there are high level of misclassification is shown in Figure 4.13.
Figure 4.23: Heatmap of posterior similarity matrix for clustering results from DPMM in case 3 of experiments when there is moderate level of misclassification injected to the training data of phoneme dataset.

The results for average estimates of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ are given in Table 4.5 and the results for prediction using test data are given in Table 4.6. In this experiment where there are high level of misclassification, a probit regression naively fitted to the observed training data (with misclassified responses) yields estimated regression coefficients: $\hat{\beta} = (-0.400, -0.224, -0.147, 0.213, 0.316, 0.253)^T$. Like the case with low level and moderate level of misclassification, the MEL perform worse in estimating the probit regression coefficients compared to the naively fitting a probit regression model to the observed training data, but all other methods provide better estimates. For prediction, the differences among various methods are not too different like in previous settings with low and moderate level of misclassification. DP-PMM seems doing slightly worse in terms of accuracy and F-measure at 0.5 threshold but does well in terms of ROC and log loss predictive probabilities.
Table 4.5: Comparison of the average estimates of $\beta_0$, $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$, $\beta_5$ in phoneme dataset under high misclassification setting where $(\bar{\gamma}_0, \bar{\gamma}_1) = (0.132, 0.169)$. The result from DP-PMM refers to the mean of estimated $\beta$ over 5000 MCMC runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>est. $\beta_0$</th>
<th>est. $\beta_1$</th>
<th>est. $\beta_2$</th>
<th>est. $\beta_3$</th>
<th>est. $\beta_4$</th>
<th>est. $\beta_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>-0.609</td>
<td>-0.387</td>
<td>-0.247</td>
<td>0.400</td>
<td>0.465</td>
<td>0.312</td>
</tr>
<tr>
<td>EM</td>
<td>-0.623</td>
<td>-0.307</td>
<td>-0.241</td>
<td>0.367</td>
<td>0.538</td>
<td>0.439</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>-0.607</td>
<td>-0.353</td>
<td>-0.232</td>
<td>0.336</td>
<td>0.498</td>
<td>0.399</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>-0.662</td>
<td>-0.367</td>
<td>-0.212</td>
<td>0.331</td>
<td>0.478</td>
<td>0.366</td>
</tr>
<tr>
<td>MEL</td>
<td>-0.313</td>
<td>-0.137</td>
<td>-0.095</td>
<td>0.133</td>
<td>0.209</td>
<td>0.168</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>-0.400</td>
<td>-0.224</td>
<td>-0.147</td>
<td>0.213</td>
<td>0.316</td>
<td>0.253</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>-0.543</td>
<td>-0.323</td>
<td>-0.287</td>
<td>0.272</td>
<td>0.479</td>
<td>0.248</td>
</tr>
</tbody>
</table>

Table 4.6: Comparison of the average estimates of accuracy, AUC, F-measure, and average log loss of predictive probability in phoneme dataset under high misclassification setting where $(\bar{\gamma}_0, \bar{\gamma}_1) = (0.132, 0.169)$. Both accuracy and F-measure are obtained at 0.5 threshold. The results from DP-PMM are the average values from using the estimated $\beta$ in every 200 run out of 5000 MCMC runs in total.

<table>
<thead>
<tr>
<th>Method</th>
<th>accuracy</th>
<th>ROC</th>
<th>F-measure</th>
<th>log loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLM to T</td>
<td>0.823</td>
<td>0.756</td>
<td>0.507</td>
<td>0.458</td>
</tr>
<tr>
<td>EM</td>
<td>0.823</td>
<td>0.756</td>
<td>0.550</td>
<td>0.459</td>
</tr>
<tr>
<td>Adjust coeff</td>
<td>0.823</td>
<td>0.757</td>
<td>0.504</td>
<td>0.459</td>
</tr>
<tr>
<td>MC-SIMEX</td>
<td>0.823</td>
<td>0.745</td>
<td>0.455</td>
<td>0.460</td>
</tr>
<tr>
<td>MEL</td>
<td>0.823</td>
<td>0.743</td>
<td>0.425</td>
<td>0.461</td>
</tr>
<tr>
<td>GLM to Y</td>
<td>0.823</td>
<td>0.749</td>
<td>0.477</td>
<td>0.486</td>
</tr>
<tr>
<td>DP-PMM</td>
<td>0.809</td>
<td>0.765</td>
<td>0.432</td>
<td>0.463</td>
</tr>
</tbody>
</table>

4.3.5 Further comparison of methods

In Section 4.3, we presented results from our experiments and discussed how they compared to true values in each misclassification setting. In this section, we discuss more on the overall performance of DP-PMM and how it compares to other methods in for modelling under misclassification.

In overall, we see from Figure 4.10, 4.15, 4.20, DP-PMM works well in estimating the regression coefficients for probit regression model for true responses. Under all three different
level of misclassification setting, the 95% credible interval contain the true probit regression coefficients. In the low misclassification setting, the DP-PMM method seem to outperform the EM method, the post-fit adjustment of regression coefficients, MC-SIMEX, and maximum estimated likelihood, giving closer estimates to the true probit regression coefficients as shown in Table 1.3. As the level of misclassification increase, all methods give estimates of regression coefficients farther from the true values. It is difficult to directly compare DP-PMM to other methods as each MCMC iteration give one set of estimates while all other methods output one set of estimates for one misclassification setting.

In term of predictive ability, MEL performs relatively worse among all the methods but all methods yield small differences in prediction outputs, especially for accuracy and F-measure. This could be due to the fact that we get the same ranking of observations being 1 versus 0 from the six methods by using AUC, accuracy, and F-measure as evaluation criteria. For accuracy and F-measure, we also have one set of results for each threshold and we display the values at 0.5 threshold in our tables. Furthermore, as discussed in Section 4.2.2, AUC is the area under ROC curve and if we examine individual ROC curves across different methods in a particular experiment run, we would see that they are all largely indifferent. Similar results has also been reported in (Rantalainen and Holmes, 2011). Regarding the log loss of predictive probability, we do see higher difference among the methods as the level of misclassification increases.

In Table 4.6, it might seem unusual that DP-PMM has lower accuracy and F-measure compared to GLM to Y despite of having estimated regression coefficient closer to the true values than those methods. However, the relatively good accuracy for GLM to Y could be because they produce the same ranking of observations as explained earlier and could be due to the imbalance in the labels in test set. It is possible that DP-PMM produced different decision boundary where it correctly classify the 0’s in the observations that are harder to
predict. Furthermore, the accuracy and F-measure we reported in Table 4.6 are for 0.5 threshold. This is likely not a good threshold for discriminating observations in DP-PMM as we can observe high value of AUC by DP-PMM in Table 4.6 when the AUC for all other methods are lower and similar. Generally speaking, DP-PMM perform well for prediction for unseen test data in terms all metrics and is comparable to EM and the post-fit adjustment of regression coefficients.

There are multiple advantages of DP-PMM compared to other methods. First, DP-PMM does not assume known non-differential misclassification probabilities and allows nonparametric estimation of differential misclassification probabilities without assuming the availability of validation data. In cases where additional information about specificity and specificity of observed responses are available, DP-PMM can also easily incorporate them through the choice of hyper-priors. Secondly, DP-PMM provide inference on the misclassification probabilities, which could be of interest and can be used to check the sensibility of the learned model. Posterior distribution of the probit regression parameters are also available as opposed to point estimates. The true responses are also generated in the sampling process and could be a useful by-product for practitioners.

4.3.6 Further consideration on DP-PMM

In DP-PMM, it is hard to assess how many cluster we should expect. One question one might ask is how much of the clustering in DP-PMM is due to the structure in the covariates and how much is due to the misclassification probabilities. We investigate this question by checking the number of nonempty clusters when the misclassification probabilities are absent. Figure 4.24 visualises the number of nonempty clusters among 5000 MCMC runs when there is no misclassification injected and misclassification probabilities are set to be 0 for all observations. This looks very similar to Figure 4.6 where there is also no misclassification.
injected but misclassification probabilities are learned from DP-PMM and can vary.

Figure 4.24: Histogram of number of nonempty clusters among 5000 MCMC runs when there is no misclassification injected in the training data of phoneme dataset and misclassification probabilities are set to be 0 for all observations.

Thea mean number of clusters is 28 in Figure 4.6 and the histogram for number of clusters looks similar in all three settings when different levels of misclassification are added. When there are misclassification injected to the phoneme training data, the number of clusters also do not vary much with mean number of clusters being 29 for low misclassification setting, 31 for moderate misclassification setting and 32 for high misclassification setting.

Though the number of clusters seem to be not too different when the misclassification probabilities vary, the clustering results do seem to differ. Looking at the heatmaps of posterior similarity matrix for clustering results in Figure 4.13, Figure 4.18, and Figure 4.23, we can see that the size of clusters and cluster memberships of observations do vary significantly in each case.
Chapter 5

Final remarks

In this thesis we explored a variety of methods for analysing binary data under the presence of response misclassification. We described them in groups depending on whether they treat misclassification probabilities as known or unknown.

Assuming known misclassification, Neuhaus (1999) approximates the bias of an assumed model through Taylor series and subsequently use it to adjust the coefficients of the model directly fitted to observed data. (Magder and Hughes, 1997) uses EM to iteratively estimate parameters of a specified model. Rousseeuw and Christmann (2003) maximises an alternative likelihood incorporating the misclassification probabilities and Küchenhoff et al. (2006) simulates more data with different amount of misclassification to learn the effects of misclassification on model parameters for a pre-specified model and therefore attain estimates of the model parameters for that pre-specified model in the absence of misclassification. When the level of misclassification is unknown, distributional assumptions about the misclassification probabilities are needed or validation data which can be used to learn the misclassification probabilities. (Lyles et al., 2011) uses the validation data to learn misclassification probabilities and (Edwards et al., 2013) uses the validation data to impute true responses, which are treated as missing. Katz and Katz (2010) resembles the Bayesian version of (Lyles et al.,
in which the misclassification probabilities and GLM model parameters are modelled jointly with prior restrictions put on the parameters that model the misclassification probabilities. The use of validation data has the advantage of avoiding distributional assumption of $\gamma_0$, $\gamma_1$, which would likely yield more flexible and realistic modelling of misclassification probabilities. However, validation data is rarely available and often costly to obtain.

To model misclassification probabilities across observations in a flexible way without assuming the availability of validation data, we propose Dirichlet process probit misclassification mixture model for mislabelled binary data. The main idea of DP-PMM is to assume a Dirichlet process mixture model over the covariate space and misclassification probabilities. Our DP-PMM can model differential misclassification without assuming the availability of validation data. We develop an efficient algorithm for DP-PMM motivated by the truncated Gibbs sampler of Ishwaran and James (2001) that approximates the infinite DP with a finite components and updates the posterior via Gibbs sampling. DP-PMM outputs posterior distribution of component-specific parameters including the misclassification probabilities in addition to posterior distribution of the regression coefficients of probit regression model. If prior information about the misclassification probabilities are available, they can also be incorporated in DP-PMM through the choice of priors.

5.1 Future directions

To extend our work further, there are multiple venues worth considering.

5.1.1 Extend the link function beyond probit link

A natural direction for further research of our Dirichlet process probit misclassification mixture model is to allow link functions other than the probit link. The current probit regression model assumption for true responses is motivated by simple posterior Gibbs sampling of pro-
bit Bayesian regression model. Logit link, for example, is another widely-used link function for modelling binary data, partially because of the interpretation of modelling the log odds of an event. However, the posterior sampling of Bayesian logistic regression is not straightforward due to the intractability of the posterior. One way to resolve this problem is to approximate the multivariate logistic with a multivariate $t$ distribution in the posterior sampling and use a combination of Gibbs update and Metropolis-Hasting sampling (Albert and Chib, 1993) (see more details in Appendix E.3). Another possible avenue is to use the Pólya Gamma scheme proposed in Polson et al. (2013). One direction for future work is to revise of our current DP-PMM to account a logistic regression model for the true responses.

One shortcoming of the probit link that we used, which also applies to logit link, is that it is a symmetric link function and can be inappropriate for imbalance data (Yin et al., 2020). With imbalanced data, skewed link functions such as generalised extreme value, skewed Weibull and Frechet link proposed by Yin et al. (2020) can be used.

### 5.1.2 Automatic misclassification type selection

Our Dirichlet process probit misclassification mixture model naturally models non-differential misclassification. One interesting direction of future research is to have automatic selection of misclassification type. The model would need to quantify whether there is differential misclassification in the data and collapse to non-differential misclassification if they are not differential. One potential direction for this is to use enriched Dirichlet process prior proposed by Wade et al. (2014).

To quantify whether there is differential misclassification, we can get a distribution for the misclassification probability for each observation $i$ by looking up the $\gamma_{ih}^{(t)}$, $\gamma_{ih}^{(t)}$ in cluster $h$ where $z_{i}^{(t)} = h$ for each observation in every iteration $t$. In the cases where the misclassification probabilities are not differential, there is no need for mixture modelling and we
would just have two parameters $\gamma_0, \gamma_1$ for misclassification probabilities. To have a model that select misclassification type automatically, more thoughts need to put into how to allow all $\gamma_{0h}$'s and $\gamma_{1h}$'s to be the same through some variable selection process.

5.1.3 Incorporation of validation data

Another interesting direction of research is how to incorporate validation data in DP-PMM if they are available. The simplest way to incorporate validation data is to use the $t_i, y_i$ in the validation data to get an informative prior on $\gamma_{0h}, \gamma_{1h}$ to fed into DP-PMM. However, this simple incorporation does not make use of the covariates in the validation data. There are likely multiple others ways to incorporate validation data in DP-PMM.

One possibility is to incorporate validation data in the likelihood like the fully Bayesian parametric modelling method of Katz and Katz (2010) discussed in Section 1.4.2. The likelihood for data with only $x_i$ and $y_i$ can be written in (1.25) whereas the likelihood for data with also $t_i$ takes a different form as shown in (1.26).

Another possible direction to use the validation data is to adopt a cluster-specific prior for $\gamma_{0h}, \gamma_{1h}$. In (3.7), instead of updating the $\gamma_{0h}^{(t)}, \gamma_{1h}^{(t)}$ based on pre-determined prior parameters $a_0, b_0, a_1, b_1$ where $\gamma_{0h} \sim Beta(a_0, b_0), \gamma_{1h} \sim Beta(a_1, b_1)$, the prior parameters $a_0, b_0, a_1, b_1$ should take in consideration of the counts of false negatives, true positives, false positives, and true negatives from the validation data in cluster $h$.

5.1.4 Applications

Our Dirichlet process probit misclassification mixture model provides a flexible way to model mislabelled binary data. It is not limited by application and can be used to any binary dataset with continuous covariates where $X$ can be assumed to be a mixture of Gaussians. We expect DP-PMM to work especially well when the data naturally heterogeneity in mis-
classification of responses. This could be an aggregated dataset where the responses are prone to different level of errors or some data where the responses reflect some group effects and individual effects. For example, in a medical dataset where the disease outcomes are measured by different tests, we expect DP-PMM to perform well in picking up the specificity and sensitivity of each test.
Appendix A

Supplementary material for methods from literature review

A.1 Derivations of the magnitude of bias

Derivation of the magnitude of bias for any GLM

Expanding on equation (1.11),

\[ H'(x_i) = g'\{(1 - \gamma_1 - \gamma_0)g\{\beta_0 + x_i^j\beta\} + \gamma_0\} \times (1 - \gamma_1 - \gamma_0) \frac{x_i^j}{g'\{g(\beta_0 + x_i^j\beta)\}} \]

\[ - g'\{(1 - \gamma_1 - \gamma_0)g\{\beta_0 + x_i\beta\} + \gamma_0\} \times (1 - \gamma_1 - \gamma_0) \frac{x_i}{g'\{g(\beta_0 + x_i\beta)\}}. \]

Plug in \( x_i = 0 \),

\[ H'(0) = g'\{(1 - \gamma_1 - \gamma_0)g\{\beta_0 + \beta\} + \gamma_0\} \times (1 - \gamma_1 - \gamma_0) \frac{1}{g'\{g(\beta_0 + \beta)\}} \]

\[ = \frac{(1 - \gamma_1 - \gamma_0)g'\{(1 - \gamma_1 - \gamma_0)g\{\beta_0\} + \gamma_0\}}{g'\{g(\beta_0)\}}. \]
since the Taylor series is expanded at $\beta = 0$.

**Derivation of the magnitude of bias $B_{LR}$ for logistic regression**

For logistic regression, $y = g(x) = \log \left( \frac{x}{1-x} \right)$, thus

$$x = g(y) = \frac{e^y}{1 + e^y} \quad \text{and} \quad g'(x) = \frac{1}{x(x-1)}.$$  

Therefore, the denominator of (1.13) can be rewritten as

$$g'\{g(\beta_0)\} = \left\{ \frac{e^{\beta_0}}{1 + e^{\beta_0}} \times \left( \frac{e^{\beta_0}}{1 + e^{\beta_0}} - 1 \right) \right\}^{-1} = -\frac{(1 + e^{\beta_0})^2}{e^{\beta_0}}.$$  

The numerator can be simplified to

$$g'\{(1 - \gamma_1 - \gamma_0)g(\beta_0) + \gamma_0\} = g'\left\{ \frac{(1 - \gamma_1 - \gamma_0)e^{\beta_0}}{1 + e^{\beta_0}} + \gamma_0 \right\}$$  

$$= g'\left\{ \frac{e^{\beta_0} - \gamma_1 e^{\beta_0} + \gamma_0}{1 + e^{\beta_0}} \right\}$$  

$$= \left( \frac{e^{\beta_0} - \gamma_1 e^{\beta_0} + \gamma_0}{1 + e^{\beta_0}} \times \frac{-1 - \gamma_1 e^{\beta_0} + \gamma_0}{1 + e^{\beta_0}} \right)^{-1}$$  

$$= \frac{(1 + e^{\beta_0})^2}{(e^{\beta_0} - \gamma_1 e^{\beta_0} + \gamma_0)(-1 - \gamma_1 e^{\beta_0} + \gamma_0)}.$$  

Combining these two expressions, the bias factor of logistic regression can be written as

$$H'_{LR}(0) = \frac{-(1 - \gamma_1 - \gamma_0)e^{\beta_0}}{(e^{\beta_0} - \gamma_1 e^{\beta_0} + \gamma_0)(-1 - \gamma_1 e^{\beta_0} + \gamma_0)}$$  

$$= \frac{(1 - \gamma_1 - \gamma_0)e^{\beta_0}}{(e^{\beta_0}(1 - \gamma_1) + \gamma_0)\{\gamma_1 e^{\beta_0} + 1 - \gamma_0\}}$$

as shown in equation (1.14). Note that the magnitude of bias $H'(0)$ in (1.13) is a function of
\( \gamma_0, \gamma_1 \) and \( \beta_0 \). The parameter \( \beta_0 \) is the intercept term of the correct model for true responses and is generally not known but used directly in the simulation study in Neuhaus (1999) and our experiments.

### A.2 Adding misclassification in MC-SIMEX

As discussed in Section 1.3.3, MC-SIMEX calls for generating additional datasets with added misclassification by raising power of the original misclassification matrix \( \Pi \). The reason that raising the power of \( \Pi \) corresponds to adding more contamination in the responses lies behind the eigen-decomposition \( \Pi = E^{-1}\Lambda E \) and the proof is outlined below.

**Proof.** To obtain the eigenvalues of \( \Pi \), its determinant are first calculated where

\[
\det \begin{bmatrix}
1 - \gamma_0 - x & \gamma_0 \\
\gamma_1 & 1 - \gamma_1 - x
\end{bmatrix} = (1 - \gamma_0)(1 - \gamma_1) - [(1 - \gamma_0) + (1 - \gamma_1)]x + x^2 - \gamma_0\gamma_1
\]

\[
= x^2 - (2 - \gamma_0 - \gamma_1)x + 1 - \gamma_0 - \gamma_1
\]

\[
= [x - (1 - \gamma_0 - \gamma_1)](x - 1).
\]

Therefore, the eigenvalues of \( \Pi \) are 1 and \( 1 - \gamma_0 - \gamma_1 \). Plugging these two values in, the following eigenvectors (columns of \( E \)) can be found where \( \Pi = E^{-1}\Lambda E \) with

\[
\Lambda = \begin{bmatrix}
1 & 0 \\
0 & 1 - \gamma_0 - \gamma_1
\end{bmatrix}, \quad E = \begin{bmatrix}
1 & \gamma_0 \\
1 & -\gamma_1
\end{bmatrix}, \quad E^{-1} = \frac{1}{\gamma_0 + \gamma_1} \begin{bmatrix}
\gamma_1 & \gamma_0 \\
1 & -1
\end{bmatrix}.
\]
Letting \( A = (1 - \gamma_0 - \gamma_1)^\lambda \), the powers of \( \Pi \) can be written as

\[
\Pi^\lambda = E^{-1} A^\lambda E = \frac{1}{\gamma_0 + \gamma_1} \begin{bmatrix}
\gamma_1 & \gamma_0 \\
1 & -1
\end{bmatrix} \begin{bmatrix}
1 & 0 \\
0 & A
\end{bmatrix} \begin{bmatrix}
1 & \gamma_0 \\
1 & -\gamma_1
\end{bmatrix}
\]

\[
= \frac{1}{\gamma_0 + \gamma_1} \begin{bmatrix}
\gamma_1 + \gamma_0 A & \gamma_0 \\
1 & -A
\end{bmatrix} \begin{bmatrix}
1 & \gamma_0 \\
1 & -\gamma_1
\end{bmatrix}
\]

\[
= \frac{1}{\gamma_0 + \gamma_1} \begin{bmatrix}
\gamma_1 A + \gamma_0 & \gamma_0 \gamma_1 (1 - A) \\
(1 - A) & \gamma_0 A + \gamma_1
\end{bmatrix}
\]

As \( \lambda \) increases, \( 0 < A < 1 \) gets smaller and \( 0 < 1 - A < 1 \) gets bigger. Thus the off-diagonal terms \( \gamma_0 \gamma_1 (1 - A) \) and \( (1 - A) \) both get larger, implying more misclassification as \( \lambda \) increases.

For \( \Pi^\lambda \) to be well-defined, \( \lambda \geq 0 \) is needed and \( \Pi^\lambda \) must exist and represent a sensible misclassification matrix. For binary classification, these constraints translate to \( \det(\Pi) = (1 - \gamma_0) + (1 - \gamma_1) - 1 = 1 - \gamma_0 - \gamma_1 > 0 \). This trick relates different levels of misclassification by parameter \( \lambda \) where the case with no misclassification, i.e. the naive MC-SIMEX estimator, corresponds to the case where \( \lambda = 0 \) corresponds to the level of misclassification in the observed data (Küchenhoff et al., 2006). The generation of additional datasets with added misclassification utilises the above eigen-decomposition. New misclassification matrix \( \Pi^\lambda_k \) is calculated through first calculating the eigenvalues and eigenvectors of \( \Pi \), namely \( E \) and \( V \). Then \( \Pi^\lambda_k \) is obtained from evaluating \( EA^k E \) and new responses are created by sampling the observed responses with the misclassification matrix represented by \( \Pi^\lambda_k \).
A.3 Properties of MEL estimator and computing robust distance

In this Section, we first discuss some properties of MEL estimator. Then, we describe how to compute robust distance, which is required in the weighted MEL (WEMEL). When the data is not high dimensional and all covariates are continuous, the robust distance refers to one that come out from the minimum covariance determinant (MCD) estimator of Rousseeuw and Driessen (1999). In the case that not all covariates are continuous or there are many of them, the robust distances refers to the one provided by the robust principal components algorithm (robust PCA) of Hubert et al. (2002) (Rousseeuw and Driessen, 1999).

Properties of MEL estimator

A nice property of the MEL estimator of Rousseeuw and Christmann (2003) is that it always exists and is unique. Differentiating the estimated log likelihood in (1.17) with respect to \( \beta \) yields the score function

\[
S(\beta, \beta_0|\tilde{y}_1, \ldots, \tilde{y}_n, d_1, \ldots, d_n) = \prod_{i=1}^{n} \left[ \tilde{y}_i - g^{-1}(x_i\beta + \beta_0) \right] x_i.
\]

(A.1)

The MEL estimates of \( \beta, \beta_0 \)'s are obtained by setting (A.1) to zero. The Hessian matrix of (A.1) is

\[
\frac{\partial}{\partial \beta} S(\beta) = -\prod_{i=1}^{n} g^{-1}(x_i\beta + \beta_0)[1 - g(x_i\beta + \beta_0)] x_i, x_i,
\]

(A.2)

which is negative definite because the design matrix has rank \( p \) (Rousseeuw and Christmann, 2003). Therefore, the differentiable function (A.2) is strictly concave and the uniqueness follows.
Robust distance from minimum covariance determinant

When all covariates are continuous and not high-dimensional (no more than 30 covariates), the robust distance refers to one defined in the minimum covariance determinant (MCD) estimator of (Rousseeuw and Driessen, 1999). The robust distance of \( x_i \) measures how far away \( x_i \) is from the center of all observations, relative to the size of the sample. The commonly-known Mahalanobis distance serve the same purpose and is defined by

\[
MD(x_i) = \sqrt{(x_i - \bar{x})^T S^{-1}(x_i - \bar{x})}.
\]

The robust distance from minimum covariance determinant (MCD) is defined in a similar manner but motivated to provide a robust estimate of multivariate location and scatter. The MCD estimator finds \( h \) observations out of \( n \) whose covariance matrix has the lowest determinant (Rousseeuw and Driessen, 1999). The robust distance from MCD uses the mean and covariance of MCD in its calculation, leading to

\[
RD(x_i) = \sqrt{(x_i - \hat{\mu}_{MCD})^T S_{MCD}^{-1}(x_i - \hat{\mu}_{MCD})}
\]

where \( \hat{\mu}_{MCD} \) is the MCD estimate of location and \( S_{MCD} \) is the MCD covariance estimate (Rousseeuw and Driessen, 1999).

The MCD estimator can be calculated by the an iterative scheme using relative distance. Let \( H_1 \) denote a subset of \( h \) observations, its mean and covariance matrix can be calculated and denoted by \( T_1 \) and \( S_1 \) where

\[
T_1 = \frac{1}{h} \sum_{i \in H_1} x_i, \quad S_1 = \frac{1}{h} \sum_{i \in H_1} (x_i - T_1)^T (x_i - T_1).
\]
Using $H_1$ as the subset, the relative distance of $x_i$ is defined as

$$d_1(x_i) = \sqrt{(x_i - T_1)^T S_1^{-1}(x_i - T_1)}.$$  (A.3)

The motivation to define relative distance is that it can be used to guide the selection of another subset of observations of the same size, denoted by $H_2$, such that $\det(S_2) \leq \det(S_1)$ (with equality occurring only when $T_1 = T_2$ and $S_1 = S_2$). $H_2$ is defined to be the set of $h$ observations with the smallest relative distances computed following (A.3). (The essence of the proof comes down to showing $\det(S_1) \leq \det(\lambda S_1) \leq \det(S_2)$ where $\lambda$ is defined to be $1/n_p \sum_{i \in H_2} d_1(x_i)^2$.) The step of constructing $H_2$ from $H_1$, or a new subset from an old one, called concentration step by Rousseeuw and Driessen (1999), can be applied iteratively until a minimum covariate determinant is found. This procedure of finding MCD is outlined in Algorithm 9. The algorithm terminates when $\det(S_m) = 0$ or $\det(S_m) = \det(S_{m-1})$. Since there are only finite number of size $h$ subsets and $\det(S_1) > \det(S_2) > \det(S_3) > \ldots$, the sequence is nonnegative thus must converge (Rousseeuw and Driessen, 1999).

Note that in Algorithm 9, the first subset $H_1$ is chosen randomly. This does not need to be the case. One alternative is to start with a random subset of size $(p+1)$ and calculate the determinant of its covariance matrix $S_0$. If $\det(S_0) > 0$, use this subset to calculate relative distances of all observations and select the set of $h$ observations with the smallest relative distances as $H_1$. If $\det(S_0) > 0$ for the subset of size $(p+1)$ observations, add other random observations, one at at time, until $\det(S_0) > 0$. Then, the resulting subset with positive covariance determinant can be used to calculate relative distances and select $H_1$ (Rousseeuw and Driessen, 1999).

Furthermore, as mentioned earlier, the MCD estimator is motivated to provide a robust estimate of multivariate location and scatter, it thus serves a good measure to detect outliers (Hubert and Debruyne, 2010). Algorithm 9 requires the size of subset $h$, whose selection
should be guided by the data at hand. If a dataset is suspected to have 25% outliers, it makes sense to set $h = 0.75n$. If no information is known about the amount of outliers, a good candidate for $h$ is $(n + p + 1)/2$, which gives the highest possible breakdown value for MCD (Rousseeuw and Driessen, 1999). (The breakdown value is a popular measure of robustness of an estimator against outliers, indicating the smallest fraction of contamination in a sample that cause the estimator to break down, or take values that are arbitrarily meaningless (unbounded) (Hubert and Debruyne, 2009).)

**Algorithm 9:** An iterative procedure to find the MCD of $n$ observations.

*Input:* $n$ observations $x_1, x_2, \ldots, x_n$, size of subset $h$;

*Output:* $H_m$ with the minimum covariance determinant;

1. Start with a random subset $H_1$ with $h$ observations and positive covariance determinant and calculate its mean $T_1$ and covariance matrix $S_1$;

2. while algorithm have not converged do

3. Calculate the relative distance of all observations following the current subset $H_{m-1}$;

4. Order the observations in ascending order of its relative distances and pick the $h$ observations with the smallest relative distances to be new subset $H_m$;

5. Check if the algorithm has met the criteria for convergence by evaluating whether $\det(S_m) = 0$ or $\det(S_m) = \det(S_{m-1})$.

6. end

Note that Algorithm 9 is just a procedure to obtain MCD. Rousseeuw and Driessen (1999) also propose the FAST-MCD method, intended to speed up computation for large datasets ($n > 600$). Algorithm 9 could be slow for large dataset because each time a new subset is constructed $n$ distances needed to be calculated. FAST-MCD uses a nested system of subsets to efficiently approximate the MCD estimator and adopts $(n + p + 1)/2$ as the default $h$ (Rousseeuw and Driessen, 1999). The main idea is to carry out the concentration
step (constructing a new subset based on the previous one) in two chunks. The data is divided into $k$ subsets where around 300 observations in a subset is a good size. The concentration step is first performed multiple times in each subset and then in the merged set drawn from the best solutions from each subset (Rousseeuw and Driessen, 1999).

Robust distance from reflection-based algorithm for principle component analysis

In the case that not all covariates are continuous or there too many covariates, Rousseeuw and Christmann (2003) uses the robust distances provided by the reflection-based algorithm for principle component analysis (RAPCA) of Hubert et al. (2002). The robust RA-PCA is a variation of the principle component algorithm developed for datasets with high dimensions. Its basic principle is to first center the data around their spatial median and then move to stepwise search for orthogonal directions by means of reflection (Hubert et al., 2002).

For observation $x_i$, its robust distance from RAPCA is defined as

$$RD(x_i) = \sqrt{\sum_{j=1}^{k} \left( \frac{t_{ij} R}{s_j R} \right)^2}$$

where $t_{ij}$ corresponds to an element in the score matrix and $s_j R$ is the square root of the $j^{th}$ eigenvalue obtained by the robust RAPCA method.

A.4 Multiple imputation and Rubin’s rule

In Section 1.4.3 we described how Edwards et al. (2013) uses multiple imputation to obtain a model for the unobserved true responses. As stated earlier, MI is a general approach to the problem of missing data and its procedure consists of three steps - imputation, analysis, and pooling. The idea of imputation is to make a guess of each missing value and replace the missing values with the guessed ones. Multiple imputation, as the name suggests, is just
doing the imputation procedure multiple times to create different plausible imputed datasets. The key motivation to use MI is that a single imputation cannot reflect sampling variability under one model for non-response or uncertainty about the correct model for non-response (Little and Rubin, 2014).

Let $Q$ denote the estimand of interest and continue using $y$ denote the observed variable and $t$ the unknown variable, the idea of MI is based on the decomposition where

$$p(Q|Y) = \int p(Q | t, y) p(t | y) dy.$$  

(A.4)

The idea is to draw imputations for $t$ first through modelling $p(t | y)$ and then proceed to use $p(Q | y, t)$ to calculate the quantity of interest $Q$ from hypothetically complete data $(y, t)$ (Van Buuren, 2018).

Note that equation (A.4) implies that the actual posterior distribution of $Q$ is equal to the average over the repeated draws of $Q$. In addition, we know that

$$\mathbb{E}(Q | y) = \mathbb{E}[\mathbb{E}(Q | y, t) | y].$$

Hence, let $\hat{Q}$ denote an approximation of $Q$ (analogously, $\hat{Q}_k$ denotes the approximation of $Q$ from the $k$th imputation), the average $\hat{Q}$ from our samples is unbiased, or in other words, equal to $Q$. Therefore, a point estimate $\hat{Q}_K$ can be obtained by taking the average estimates of $Q$ from each imputation where

$$\hat{Q}_K = \frac{1}{K} \sum_{k=1}^{K} \hat{Q}_k.$$ 

Among the $K$ complete-data estimates, the standard unbiased estimate of the variance
among them is

\[ B_K = \frac{1}{K-1} \sum_{k=1}^{K} (\bar{Q}_k - Q)(\bar{Q}_k - Q)^T. \]

Rubin’s rule essentially says that the total variance \( T_K \) can be estimated by

\[ T_K = \bar{U}_K + \left( 1 + \frac{1}{K} \right) B_K, \tag{A.5} \]

where \( \bar{U}_K \) denotes the average of complete data variance and can be calculated via

\[ \bar{U}_K = \frac{1}{K} \sum_{k=1}^{K} \hat{U}_k \]

where \( \hat{U}_k \) denote the variance from the \( k \)th imputation. Rubin’s rule is motivated by the variance decomposition equation where

\[ Var(Q \mid y) = E[Var(Q \mid t, y) \mid y] + Var(E(Q \mid t, y) \mid y], \]

and Monte Carlo estimates of relevant quantities where

\[ \bar{Q}_K \approx E[E(Q \mid t, y) \mid y] = E(Q \mid y), \]
\[ \bar{U}_K \approx E[Var(Q \mid t, y) \mid y], \]
\[ (1 + 1/K)B_K \approx Var[E(Q \mid t, y) \mid y], \]
\[ T_K \approx Var[E(Q \mid y)]. \]

Note that the \( 1/K \) part arises in (A.5) because \( \bar{Q} \) itself is estimated using finite \( K \) imputations and only approximates \( \bar{Q}_\infty \) (Van Buuren, 2018).

Note that we only briefly mentioned missing data mechanisms in Section 1.1, which categorises the types of missing data into three types - missing completely at random (MCAR),
missing at random (MAR), and missing not at random (MNAR). The idea of missing mechanism is to describes the underlying mechanism that generates missing data by introducing and assigning distribution of missing-data indicators. Using $D^{mis}$ for missing data, $D$ for complete data, and $D^{obs}$ for observed data, MAR means

$$p(D^{mis}|D, \phi) = p(D^{mis}|D^{obs}, \phi).$$

Informally speaking, MAR implies that the missingness only relate to the observed data, MCAR means that the occurrence of missing values is completely at random, and NMAR refers to the case that the missing values are related to both observed and unobserved variable and the missing mechanism cannot be ignored. It is important to bear in mind that MI is designed for data with MAR missingness. In other words, an underlying assumption of MI is that the distribution of the missing data is independent of the missing values themselves, therefore can be inferred from just the observed data.
Appendix B

Existence of maximum likelihood estimator in logistic regression

B.1 Conditions for the existence of maximum likelihood estimator in logistic regression models

It is known that for LR, when the sample data points are completely separable, there are non-unique infinite estimates to the maximum likelihood. Obviously, these are not the desired solution and this problem regarding the existence, uniqueness and location of the maximum likelihood estimate was formalised in Albert and Anderson (1984) for multinomial LR models. We provide a summary of their results in this section, focusing on binary LR models. In short, the main contributions of Albert and Anderson (1984) are the partition of the pattern of data points into three group (complete separation, quasi-complete separation, and overlap) and proof regarding the existence of unique maximum likelihood estimate based these three partitions.

We use $C_1, C_2$ to denote the two classes and $E_1, E_2$ to denote their row identifies, which
indicate the observations that belong to a particular class. All observations are either in $C_1$ or $C_2$, thus $E_1 \cup E_2 = \{1, 2, \ldots, n\}$. We also use $\beta_s = (\beta_{s0}, \beta_{s1}, \ldots, \beta_{sp})$ represents the parameters characterizing the probability that an observation is in class $s$ where $s = \{1, 2\}$. $\beta = (\beta_1, \beta_2)$ represents the set of all parameters, which amounts to $v = 2(p + 1)$ unknown parameters.

The logistic approach to regression and discrimination assumes that the conditional probabilities have the extended logistic form where

$$p(1|x_i \in C_s|x_i) = \frac{\exp(x_i \beta_s)}{1 + \exp(x_i \beta_s)}, \quad s = \{1, 2\}.$$  

The discrimination rule assigns $x_i$ to $C_1$ if and only if

$$x_i (\beta_1 - \beta_2) \geq 0. \quad (B.1)$$

The log-likelihood that needs to be maximised is

$$\log L(X, \beta) = \sum_{i \in E_1} \log \left[ \frac{1}{\exp(x_i (\beta_1 - \beta_2))} \right] + \sum_{i \in E_2} \log \left[ \frac{1}{\exp(x_i (\beta_2 - \beta_1))} \right]. \quad (B.2)$$

(Albert and Anderson, 1984) suggests that the existence of finite maximum likelihood estimates of a LR depends on the sample data pattern and categorises all possible configurations of the $n$ sample data points in the observation space $R^p$ into three cases.

**Case 1: complete separation:**

For binary data, complete separation implies that for all $i \in E_1$, $x_i$ satisfies $\beta_1 x_i > 0$ and for all $i \in E_2$, $x_i$ satisfies $\beta_1 x_i < 0$. In other words, complete separation implies that there exist a $\beta \in R^p$ that correctly classify all samples into their group. An illustration of the case of complete separation when there are only two covariates and two classes is shown in the top
Theorem 1. If there is complete separation of the data points, then the $\hat{\beta}_{\text{MLE}}$ does not exist and
\[ \max_{\beta \in \mathbb{R}^p} \log L(X, \alpha) = -\infty \quad (\text{or} \quad \max_{\beta \in \mathbb{R}^p} L(X, \alpha) = 1). \]

Proof. To see this, first let $A^c$ to denote the collection of all $\beta$’s that satisfy (B.1). For $\beta \in A^c$, consider $\beta^k = k\beta$ where $k > 0$, the log-likelihood function of $\log L(X, \beta^k)$ is
\[ \log L(X, \beta^k) = \sum_{i \in E_1} \log \left[ \frac{1}{\exp\{x_i k(\beta_1 - \beta_2)\}} \right] + \sum_{i \in E_2} \log \left[ \frac{1}{\exp\{x_i k(\beta_2 - \beta_1)\}} \right] \]

For any $k > 0$, condition (B.1) holds and, as $k$ gets larger, the log likelihood $\log L(X, \beta^k)$ gets smaller. When $k$ approaches infinity, the log likelihood $\log L(X, \beta^k)$ approaches zero. Therefore, the maximum likelihood estimate when sample data has complete separation is attained at infinity.

Case 2: quasi-complete separation:
$\beta \in \mathbb{R}^n$ gives quasi-complete separation of the sample data points if
\[ x_i(\beta_1 - \beta_2) \geq 0 \quad \text{for all} \ i \in E_1, \ \text{and} \]
\[ x_i(\beta_1 - \beta_2) = 0 \quad \text{for at least one} \ i. \quad \text{(B.3)} \]

For that point $x_i$ in which $x_i(\beta_1 - \beta_2) = 0$, it is said to be quasi-separated with respect to $\beta$.

We further assume there are $r$ quasi-separated points and use $X^q$ to denote their corresponding model matrix, which is of dimension $r \times (p + 1)$. We use $A^q$ to denote the set of all $\beta$ that satisfy the equality in (B.3) and refer to the set of indexes of all the quasi-separated points as $E_\beta$. 

173
**Corollary 1.** For any dataset $X$ for which $A^q$ is non-empty, $A^q$ is a convex cone.

This follows from the fact that, first, if $\beta$ satisfies equality in (B.3) for at least one $i$, so does $k\beta$. If $\beta^1, \beta^2 \in A^q$, then $\beta' = a\beta^1 + b\beta^2$ where $a + b = 1$ is also in $A^q$. $\beta'$ satisfies the inequality in (B.3) by algebra and also holds the equality in (B.3) for at least one $i$ triplet to avoid being able to give complete separation to all data points (which leads to a contradiction).

**Corollary 2.** For any dataset $X$ for which $A^q$ is non-empty, if there a minimal set $E^m$ such that $E^m \subset E_{\beta}$ for all $A^q$, and $A^{mq} = \{\beta : \beta \in A^q \text{ and } E_{\beta} = E^m\}$ is non-empty.

In the above notation, let $i \in E_{\beta}$, then $x_i$ satisfies

$$ax_i(\beta^1_1 - \beta^1_2) + bx_i(\beta^2_1 - \beta^2_2) = 0$$

for at least one $i \in E_1$. Since $a, b > 0$ and $\beta^1, \beta^2 \in A^q$, each of the two terms in the above equation are nonnegative. Therefore, the only way for the equality to hold is that both terms are zero and $i \in E_{\beta^1}$ and $i \in E_{\beta^2}$. This implies that

$$E_{\beta'} = E_{\beta^1} \cap E_{\beta^2}.$$ 

What we have showed effectively is $E^q = \{E_{\beta} : \beta \in A^q\}$ is closed under intersection. Define

$$E^m = \cup_{\beta \in A^q} E_{\beta},$$

then by the closure property and the finiteness of each $E_{\beta}$, there is $\beta \in A^q$, $E_{\beta} = E^m \in E^q$ and $A^{mq}$ is not empty.

**Theorem 2.** If there is quasi-separation of the data points, then the $\hat{\beta}_{MLE}$ does not exist.
and
\[
\max_{\beta \in \mathbb{R}^v} \log L(\mathbf{X}, \beta) = \max_{\beta' \in \mathbb{Q}^n} \log L(\mathbf{X}^{mq}, \beta') < 1
\]
where \( \mathbf{X}^{mq} \) is the matrix of quasi-separated points corresponding to the minimal set \( E^m \).

**Proof.** For any \( \beta \in \mathbb{R}^v \), we can find a nearest element \( \beta_q \in \mathbb{A}^q \) and write \( \beta = \beta_{nq} + \beta_q \) where \( \beta_{nq} \in \mathbb{A}^q \). Consider \( \beta_i = \beta_{nq} + k\beta_q \) \((k > 0)\), we will show that the likelihood function is increasing with \( k \). Let \( \tilde{E}_{\beta_q} \) denote the set of indexes of \( x_i \) that satisfy (B.3) with strict inequality (i.e. not quasi-completely separated) and \( E_{\beta_q} \) denote those that are quasi-completely separated, the log likelihood function can be written as

\[
\log L(\mathbf{X}, \beta_k) = \sum_{i \in E_{\beta_q}} \log L_i + \sum_{i \in \tilde{E}_{\beta_q}} \log L_i
\]

where
\[
L_i = \frac{1}{\sum_{i \in E_{\beta_q}} \exp[\mathbf{x}_i(\beta_1^{mq} - \beta_2^{mq}) + \mathbf{x}_i(\beta_1^q - \beta_2^q)]}.
\]

For fixed \( \beta \), \( L_i \) is monotone increasing function of \( k \) for \( i \in E_{\beta_q} \) and so is \( l(E_{\beta_q}) = \sum_{i \in E_{\beta_q}} \log L_i \). The exact same argument holds for \( i \in \tilde{E}_{\beta_q} \). Therefore, \( \log L(\mathbf{X}, \beta_k) \) is an increasing function of \( k \) for every \( \beta \in \mathbb{R}^v \) and the \( \beta_{MLE} \) is at infinity on the boundary of \( \mathbb{R}^v \).

**Case 3: overlap:**

The definition of overlap of sample points is essentially the non-existence of either complete or quasi-complete separation. This means that for any \( \beta \in \mathbb{R}^v \), there exists a point \( i \in E_1 \) satisfies

\[
\mathbf{x}_i(\beta_1 - \beta_2) < 0. \quad (B.4)
\]

If there is overlap of data points, the maximum likelihood estimate \( \hat{\beta} \) exists and is unique.
The proof of this effectively follows from the fact that the log likelihood has limit $-\infty$ at infinity and is strictly concave.

![Sample data points patterns](image)

Figure B.1: Sample data points patterns - top: completely separable, overlap; bottom: two cases of quasi-separable.

Figure B.1 shows four possible configurations of sample points when the data has two covariates and the response has two classes. The x-axis represents variable 1, y-axis represents variable 2, and the sample data points are denoted by × and o for two different classes. In the top row of Figure B.1, we show the case of data (left) is complete separable and overlap (right).

### B.2 Identification issue

One of the challenges of modelling under misclassification as we mentioned in Section 1.1 is identifiability issues. When the observed responses are misclassified, regardless of the model assumption, the likelihood of the observed data in any form inherently has identification problem. In the simplest case of assuming non-differential misclassification probabilities, the
The likelihood of the observed data takes the form of

\[ L(\beta, \gamma_0, \gamma_1) = \prod_{i=1}^{n} [(1 - \gamma_0 - \gamma_1)g(x_i\beta) + \gamma_0]^{y_i} \times [1 - (1 - \gamma_0 - \gamma_1)g(x_i\beta) - \gamma_0]^{1-y_i}. \]  

(B.5)

Optimizing this likelihood directly cannot be done because the parameters might not be identifiable. The most commonly used model for binary classification is logistic regression, and for logistic regression even in the case without response misclassification, a finite maximum likelihood estimate might not exist. For a finite and unique maximum likelihood estimate to exist, the sample data points must overlap as discussed in Appendix B.1.

In the case where there are misclassification in the responses, the identifiability issue becomes more complex. Under non-differential misclassification assumption, the misclassification probabilities must satisfy \( \gamma_0 + \gamma_1 < 1 \), for which violation implies extraordinary large amount of misclassification and estimators based on equation (B.5) cannot differentiate between \((\gamma_0, \gamma_1, \beta)\) and \((1 - \gamma_0, 1 - \gamma_1, -\beta)\). This is one of the necessary conditions for identification of the logit and probit models discussed in (Hausman et al., 1998) besides the prior requirement is that the model needs to be identifiable without misclassification. The other requirement is that the link function must be non-linear since if the link function is linear, the parameters are not unique, which can be seen from

\[ \mathbb{E}(y_i|x_i) = \gamma_0 + (1 - \gamma_0 - \gamma_1)(\beta_0 + x_i\beta_1) = (\gamma_0 + \beta_0) + (1 - \gamma_0 - \gamma_1)x_i\beta_1. \]

Lifting up the non-differential misclassification assumption, the matter of identifiability becomes naturally harder since there would be more parameters in the likelihood. The general concern of infinite number of maximum likelihood solutions has been mentioned in previous works of Gaba and Winkler (1992); Joseph et al. (1995); Swartz et al. (2004).
B.3 Firth’s correction for maximum likelihood estimator

In our implementation, we used the *brglm2* package (Kosmidis, 2017) to fit logistic regression when possible, implementing the bias-reducing adjusted scores method proposed in Firth (1993). This method avoids infinite maximum likelihood estimates in logistic regression when the two classes are separable by modifying the score function and can remove the first-order term from the asymptotic bias of maximum likelihood estimates of regular parametric problems Firth (1993).

In regular problems, the maximum likelihood estimate is derived as a solution to the score equation, which is the derivative of the log likelihood.

\[ S(\beta) = \nabla l(\beta) = \nabla \log L(\beta) = 0 \]

The idea of modified score function in Firth (1993) is that the bias in \( \hat{\beta} \) can be reduced by introducing a small bias into the score function as shown in equation (B.6).

\[ S^*(\beta) = S(\beta) - i(\beta)b(\beta) \quad (B.6) \]

If the \( \hat{\beta} \) is subject to a positive bias \( b(\beta) \), then the score function is shifted downward at each point \( \beta \) by an amount \( i(\beta)b(\beta) \) where \( S'(\beta) = -i(\beta) \) represents the Fisher information matrix (or local gradient). The modified-score functions neatly corresponds to penalisation of the likelihood by Jeffreys prior for LR and the bias-reduced estimates are always finite and beneficially shrink towards the origin (Kosmidis, 2017).
Appendix C

Distribution review

C.1 Additional notes on Beta distribution

The expectation in (2.1) can be seen as follows:

\[
E[\log(x)] = \int_0^1 \frac{1}{\text{Beta}(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1} \log(x) \, dx
= \frac{1}{\text{Beta}(\alpha, \beta)} \int_0^1 \frac{\partial}{\partial \alpha} x^{\alpha-1}(1-x)^{\beta-1} \, dx
= \frac{1}{\text{Beta}(\alpha, \beta)} \frac{\partial}{\partial \alpha} \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} \, dx
= \frac{1}{\text{Beta}(\alpha, \beta)} \int_0^1 \frac{\partial \text{Beta}(\alpha, \beta)}{\partial \alpha} \, dx
= \frac{\partial \Gamma(\alpha)}{\partial \alpha} - \frac{\partial \Gamma(\alpha + \beta)}{\partial \alpha} = \psi(\alpha) - \psi(\alpha + \beta).
\]

Similarly, \(E[\log(1-x)]\) can be expressed as taking derivative of \(\log[\text{Beta}(\alpha, \beta)]\) with respect to \(\beta\) thus leading to \(\psi(\beta) - \psi(\alpha + \beta)\).
C.2 Wishart distribution

Wishart distribution is a multidimensional generalisation of Gamma distribution. It can be defined in terms of degrees of freedom $v \geq p$ and the scale matrix $S$ where $X \sim \text{Wishart}(v, S)$ implies

$$f_X(X) = \frac{|X|^{(v-p-1)/2} \exp[-\frac{1}{2} \text{tr}(S^{-1}X)]}{2^{vp/2} \Gamma_p(v/2)|S|^{v/2}}$$

with mean $\mathbb{E}(x) = vS$ and mode $= (v - p - 1)S$ where $v > p + 1$.

C.3 Normal Inverse Wishart (NIW)

The inverse Wishart distribution is the multivariate version of inverse Gamma distribution. For $d \times d$ positive definite covariance matrix $X$, $X$ follows inverse Wishart with degree of freedom $v > d - 1$ and $d \times d$ symmetric, positive-semidefinite scale matrix $S$ if

$$f_X(X|S^{-1}) = \frac{|X|^{-(v+d+1)/2} \exp[-\frac{1}{2} \text{tr}(SX^{-1})]}{|S|^{v/2}/\{2^{vd/2} \Gamma_d(v/2)\}}$$

and $\mathbb{E}(X) = \frac{S}{v-d-1}$. 

C.4 Conjugacy with MVN

Wishart and NIW are useful distribution in practice because it forms nice conjugacy with MVN. The pdf of a multivariate normal where $X \sim \text{MVN}(\mu, \Sigma)$ follows

$$f(X|\mu, \Sigma) = (2\pi)^{-p/2}(det(\Sigma))^{-1/2} \exp\left\{\frac{1}{2} (X - \mu)(\Sigma^{-1})(X - \mu)\right\}.$$
The nice hierarchy where conjugacy holds is when

\[ X|\mu, \Sigma \sim MVN(\mu, \Sigma) \]

where priors \((\mu, \Sigma) \sim NIW(\mu_0, \Lambda_0/k_0; v_0, \Lambda_0)\). In other words,

\[ \Sigma \sim W^{-1}(v_0, \Lambda_0^{-1}), \quad \mu|\Sigma \sim N(\mu_0, \Sigma/k_0). \]

The posterior distribution for mean and covariance is also NIW where

\[ \mu, \Sigma|X \sim NIW(\mu_n, \Lambda_n/k_n; v_n, \Lambda_n) \]

or

\[ \Sigma|X \sim W^{-1}(v_n, \Lambda_n^{-1}), \quad \mu|\Sigma, X \sim N(\mu_n, \Sigma/k_n) \]

with

\[
\begin{align*}
\mu_n &= \frac{k_0}{k_0 + n} \mu_0 + \frac{n}{k_0 + n} \bar{x}, \\
k_n &= k_0 + n, \\
v_n &= v_0 + n,
\end{align*}
\]

\[
\Lambda_n = \Lambda_0 + \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T + \frac{k_0 n}{k_0 + n} (\bar{x} - \mu_0)(\bar{x} - \mu_0)^T.
\]
C.4.1 Conjugacy derivation for the Normal part

Suppose that

\[ x_1, \ldots, x_n \overset{iid}{\sim} MVN(\mu, \Sigma) \]
\[ \mu \sim MVN(u_0, \Lambda_0), \]

then the full conditional distribution of \( \mu | X, \Sigma \) is

\[ \mu | X, \Sigma \sim MVN \left( \{ \Lambda_0^{-1} + n\Sigma^{-1} \} \{ \Lambda_0^{-1} u_0 + \Sigma^{-1} n\bar{x} \}, \Lambda_0^{-1} + n\Sigma^{-1} \right). \]

**Proof.** The prior for \( \mu \) is

\[ p(\mu) = (2\pi)^{-p/2} \text{det}(\Lambda_0)^{-1/2} \exp \left\{ -\frac{1}{2} (\mu - u_0) (\Lambda_0^{-1} (\mu - u_0)) \right\} \]
\[ \propto \exp \left\{ -\frac{1}{2} \left\{ \mu^{(i)} \Lambda_0^{-1} \mu - 2 \mu^{(i)} \Lambda_0^{-1} u_0 + u_0^{(i)T} \Lambda_0^{-1} u_0 \right\} \right\} \]
\[ \propto \exp \left\{ -\frac{1}{2} \left\{ \mu^{(i)} \Lambda_0^{-1} \mu - 2 \mu^{(i)} \Lambda_0^{-1} u_0 \right\} \right\} = \exp \left\{ -\frac{1}{2} \left\{ \mu^{(i)} A_0 \mu - 2 \mu^{(i)} b_0 \right\} \right\} \]

where

\[ A_0 = \Lambda_0^{-1}, \quad b_0 = \Lambda_0^{-1} u_0. \]
The likelihood for \( X = (x_1, ..., x_n) \) is

\[
p(X|\mu, \Sigma) = \sum_{i=1}^{n} (2\pi)^{-p/2} \det(\Sigma)^{-1/2} \exp \left\{ - \frac{1}{2} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) \right\}
\]

\[
\propto \exp \left\{ - \frac{1}{2} \left\{ \sum_{i=1}^{n} x_i^T \Sigma^{-1} x_i - 2 \sum_{i=1}^{n} \mu^T \Sigma^{-1} x_i + \sum_{i=1}^{n} \mu^T \Sigma^{-1} \mu \right\} \right\}
\]

\[
\propto \exp \left\{ - \frac{1}{2} \left\{ -2 \sum_{i=1}^{n} \mu^T \Sigma^{-1} x_i + \sum_{i=1}^{n} \mu^T \Sigma^{-1} \mu \right\} \right\}
\]

\[
\propto \exp \left\{ - \frac{1}{2} \left\{ -2 \mu^T \Sigma^{-1} n\bar{x} + \mu^T n\Sigma^{-1} \mu \right\} = \exp \left\{ \frac{1}{2} \left\{ -2 \mu^T b_1 + \mu^T A_1 \mu \right\} \right\}
\]

where

\[A_1 = n\Sigma^{-1}, \quad b_1 = \Sigma^{-1} n\bar{x}.
\]

Therefore, the full conditional follows

\[
\mu|X, \Sigma \propto p(X|\mu, \Sigma) \times p(\mu)
\]

\[
\propto \exp \left\{ - \frac{1}{2} \{ \mu^T A_0 \mu - 2 \mu^T b_0 \} \right\} \times \exp \left\{ \frac{1}{2} \left\{ -2 \mu^T b_1 + \mu^T A_1 \mu \right\} \right\}
\]

\[
\propto \exp \left\{ \mu^T b_1 + \mu^T b_0 - \frac{1}{2} \mu^T A_0 \mu - \frac{1}{2} \mu^T A_1 \mu \right\}
\]

\[
= \exp \left\{ - \frac{1}{2} \mu^T (A_0 + A_1) \mu + \mu^T (b_1 + b_0) \right\}
\]

which is MVN with mean \((A_0 + A_1)^{-1}(b_0 + b_1)\) and variance \(A_0 + A_1\). \(\square\)
C.4.2 Conjugacy derivation for the Inverse Wishart part

Continuing with $X|\mu, \Sigma \sim MVN(\mu, \Sigma)$, now suppose $\Sigma \sim W^{-1}(v_0, \Lambda_0^{-1})$ where $v_0$ is a scalar and $\Lambda_0^{-1}$ is a matrix, then

$$p(\Sigma) \propto \det(\Sigma)^{-(v_0+p+1)/2} \times \exp\{-tr(\Lambda_0\Sigma^{-1})/2\}.$$  

The posterior distribution follows

$$p(\Sigma|X, \mu) \propto p(\Sigma) \times p(X|\mu, \Sigma) \propto \det(\Sigma)^{-(v_0+n+1)/2} \times \exp\{-tr(\Lambda_0\Sigma^{-1})/2\} \times \det(\Sigma)^{-n/2} \times \exp\{-tr(S_\mu\Sigma^{-1})/2\} \times \det(\Sigma)^{-((v_0+n)+p+1)/2} \times \exp\{-tr((\Lambda_0 + S_\mu)\Sigma^{-1})/2\}$$

where $tr(S_\mu\Sigma^{-1}) = \sum_{i=1}^{n} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)$. This implies that the posterior of $\Sigma$ follows

$$\Sigma|X \sim W^{-1}(v_n, \Lambda_n)$$

where

$$v_n = v_0 + n \quad \Lambda_n = \Lambda_0 + \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T + \frac{k_0n}{k_0+n}(\bar{x} - \mu_0)(\bar{x} - \mu_0)^T$$

C.4.3 Sampling of normal inverse wishart distributed variable

Therefore, suppose we assume $X|\mu, \Sigma \sim MVN(\mu, \Sigma)$ where

$$\Sigma \sim W^{-1}(v_0, \Lambda_0^{-1}) \quad \mu|\Sigma \sim N(\mu_0, \Sigma/k_0),$$

184
a Gibbs sampler can be used to sample posteriors of $\mu, \Sigma$. This comes down to at iteration $k$,

1. sample $\Sigma^{(k+1)}$ from its full conditional:
   
   1.1. compute $\Lambda_n$ from $X$ and $\Sigma^{(k)}$,
   1.2. sample $\Sigma^{(k+1)} \sim W^{-1}(v_n, \Lambda_n^{-1})$,

2. sample $\mu^{(k+1)}$ from its full conditional:
   
   2.1. compute $u_n$ and $\Sigma_n$ from $X$ and $\Sigma^{(k+1)}$,
   2.2. sample $\mu^{(k+1)} \sim MVN(u_n, \Sigma_n)$.

**C.4.4 Updating of normal and inverse wishart parameter in each cluster**

In implementation, we assume that

$$
\Sigma \sim W^{-1}(d_0 - p + 1, S_0) \quad \mu|\Sigma \sim MVN(0, t_0 \Sigma)
$$

where $S_0$ is the sum of squares in prior.

Thus, following our previous notation where

$$
\Sigma \sim W^{-1}(v_0, \Lambda_0^{-1}) \quad \mu|\Sigma \sim N(\mu_0, \Sigma/k_0)
$$

we now have

$$\mu_0 = 0, \quad k_0 = t_0^{-1}, \quad v_0 = d_0 - p + 1, \quad S_0 = \Lambda_0^{-1}.$$
The posterior distribution for mean and covariance follows

$$\Sigma | X \sim W^{-1}(v_n, \Lambda_n^{-1}), \quad \mu | \Sigma, X \sim N(\mu_n, \Sigma / k_n)$$

with

$$\mu_n = \frac{n}{k_0 + n} \bar{x} = \frac{n}{t_0^{-1} + n} \bar{x}, \quad k_n = k_0 + n = t_0^{-1} + n,$$

$$v_n = v_0 + n = d_0 - p + 1 + n,$$

$$\Lambda_n = \Lambda_0 + \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T + \frac{k_0 n}{k_0 + n} (\bar{x} - u_0)(\bar{x} - u_0)^T$$

$$= S_0^{-1} + \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T + \frac{t_0^{-1} n}{t_0^{-1} + n} (\bar{x} - u_0)(\bar{x} - u_0)^T.$$

In the context of each cluster following a NIW, $n$ should be adapted to $n_k$ to reflect number of observations in cluster $k$. The observations $x$ counted should also be restricted to those observations with latent class $c_i = k$.

With $p = 2, d_0 = 5$, we have

$$E(\Sigma) = \frac{S_0}{d_0 - p + 1 - p - 1} = \frac{S_0}{d_0 - 2p} = S_0.$$

Since $\mu | \Sigma \sim MVN(0, t_0 \Sigma)$, the expected variance of of $\mu$ is

$$E(Var(\mu | \Sigma)) = E(t_0 \Sigma) = t_0 E(\Sigma) = t_0 S_0.$$

We set

$$S_0 = d_0 * \text{diag}(p)/5 = \text{diag}(p),$$

followed by $t_0 = 10$ to balance the expected variance of $\mu$. 186
Appendix D

Additional discussion on Dirichlet process and Dirichlet process mixture models

D.1 Black-MacQueen urn scheme

The Black-MacQueen urn scheme (or Pólya urn scheme) was first described in Blackwell et al. (1973). It provides a generative model for producing random measure instances specified by the DP thus proves the existence of the DP. The scheme sets around picking coloured balls from an urn and putting them back. To begin, we start with an empty urn that will be used to contain previously seen balls. Let each value in $\Theta$ represent a unique colour and draws $\theta \sim P$ denote the colour of the balls drawn. In the first draw, we pick a colour drawn from $P_0$, i.e. draw $\theta_1 \sim P_0$. We paint a ball with that colour and drop it into the urn. In subsequent steps, say at step $n + 1$, we will either with probability $\frac{\alpha}{\alpha + \eta}$ pick a new colour (draw $\theta_{n+1} \sim P_0$), paint a ball with that colour and drop it into the urn, or with probability $\frac{\eta}{\alpha + \eta}$ reach into the urn to pick a random ball out (draw $\theta_{n+1}$ from empirical distribution),
paint a ball with the same colour, and drop both balls back into the urn.

The colours \( \theta_1, \theta_2, \ldots \) we draw from the Black-MacQueen urn scheme refer to draws from \( P \sim DP(\alpha P_0) \) (Teh et al., 2005). Blackwell et al. (1973) proves the Black-MacQueen urn scheme leads to a Dirichlet process with the use of De Finetti’s theorem. The drawing process follow the conditional distribution in (2.3), where a distribution over sequences \( \theta_1, \theta_2, \ldots \) can be constructed by iteratively drawing each \( \theta_i \) given \( \theta_1, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n \) (Ishwaran and Zarepour, 2002). This random sequence is infinitely exchangeable such that for any permutation \( \sigma \) on \( 1, \ldots, n \), we have

\[
p(\theta_1, \ldots, \theta_n) = p(\theta_{\sigma(1)}, \ldots, \theta_{\sigma(n)}).
\]

De Finetti’s theorem states that for any infinitely exchangeable sequence \( \theta_1, \theta_2, \ldots \), there is a random distribution \( G \) such that the sequence is composed of i.i.d. draws from it (Aldous, 1985), expressed by

\[
P(\theta_1, \ldots, \theta_n) = \int \prod_{i=1}^{n} G(\theta_i) dp(G).
\]

That is, De Finetti’s theorem guarantees a mixture model provided that the observations are exchangeable like in the case here in Black-MacQueen urn scheme (Zhang, 2008). In this case, the prior over the random distribution \( p(G) \) is precisely the Dirichlet process \( DP(\alpha P_0) \). Therefore, in Black-MacQueen urn scheme, De Finetti’s theorem implies that there exists a prior over the random distribution \( P(G) \) and \( P(G) \) is exactly the Dirichlet process \( DP(\alpha P_0) \) (Teh, 2010).

### D.2 Chinese Restaurant Process

While the Blackwell-MacQueen urn scheme shows the existence of the Dirichlet process, the *Chinese Restaurant Process* (CRP) is a metaphor that gives a more intuitive representation
of Dirichlet Process and illustrates its key property of exchangeability. CRP starts out by
drawing $\theta_1, \ldots, \theta_n$ from the Black-MacQueen urn scheme. After $n$ balls are drawn, the values
of $\theta_1, \ldots, \theta_n$ induce a partition on $[n] = \{1, \ldots, n\}$. The Chinese Restaurant Process is this
induced distribution over partitions (Fruhwirth-Schnatter et al., 2019).

The Chinese Restaurant Process metaphor sets in a Chinese restaurant with infinite
seating capacity. The first customer enters the restaurant and sits at an initially empty
table. The second customer comes in and can either sit with the first customer or at an
unoccupied table. The process goes on, and generally speaking, the $n + 1$ customer has the
option to join any already occupied table $k$ with probability proportional to the number of
customers already sitting at this table, $n_k$, or sit at a new table with probability proportional
to the concentration parameter $\alpha$ (Aldous, 1985). In the CRP metaphor, $\theta_i$ is the table that
the $i$th customer joined and for $i \geq 1$, $\theta_i$ takes on a new value with probability $\frac{\alpha}{\alpha + i - 1}$,
independent of the number of clusters among previous $\theta_1, \ldots, \theta_{n-1}$ (or the number of already
occupied tables).

The CRP illustrates the clustering property (effect) of the Dirichlet process from another
angle. Following the original (formal) definition, Ferguson (1973) showed that the conditional
distribution of a draw follows

$$
\theta_i | \theta_1, \ldots, \theta_{i-1}, \alpha, P_0 \sim \frac{\alpha}{\alpha + i - 1} P_0 + \sum_{j=1}^{i-1} \frac{1}{\alpha + i - 1} \delta_{\theta_j}
$$

also known as the Pólya urn prediction rule (Gelman et al., 2013). In the CRP metaphor,
this means the positive reinforcement effect where the more people sit at a table, the more
likely some new customer will join that table. What matters is not which customers sit at
which table but the occupancy number at the tables. In other words, using $\theta_{-i}$ to denote all
\( \theta_j \) such that \( j \neq i \), we can rewrite the conditional distribution in (D.1) as

\[
\theta_i | \theta_{-i}, \alpha, P_0 \sim \frac{\alpha}{\alpha + n - 1} P_0 + \sum_{l=1}^{k(-i)} \frac{n_l(-i)}{\alpha + n - 1} \delta_{\theta_l(-i)}
\]  

(D.2)

where \( \theta_l^{(-i)} \) are unique values of \( \theta^{(-i)} \) and \( n_l^{(-i)} = \sum_{j \neq i} I[\theta_j = \theta_l] \) (Teh et al., 2005).

In the CRP metaphor, the clustering effect can also be seen from the expected number of unique tables (clusters). Let \( n^* \) denote the number of unique clusters, Antoniak (1974) showed that

\[
\mathbb{E}(n^*|n) = \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \approx \alpha \log \left( 1 + \frac{n}{\alpha} \right).
\]

The number of unique clusters grows logarithmically in the number of observations \( n \) but the existing non-empty clusters get bigger (thus the clustering effect is also referred to as the rich-get-richer phenomenon) (Teh, 2010).

### D.3 Marginalised Gibbs sampler

As discussed in Pitman (1996), the Dirichlet process is a special case of two-parameter Poisson-Dirichlet (Pitman-Yor) process \( \mathcal{PY}(a, b) \) where \( a = 0, b = \alpha \). The joint distributions of samples can be characterised by the generalised Pólya urn scheme where Pitman (1996) gives a prediction rule. This generalised Pólya urn prediction rule simplifies to the Pólya urn prediction rule in (D.1) for the Dirichlet process case when \( a = 0, b = \alpha \). This Gibbs sampler we discuss now is based on this prediction rule and the algorithm is often referred to as the Pólya urn Gibbs sampler or marginalised Gibbs sampler. The main idea is to resolve the problem that the mixing measure \( P \) is characterised by infinite many parameters, we marginalise out \( P \) to obtain an induced prior distribution on the subject specific parameters \( \theta_1, \ldots, \theta_n \) (Gelman et al., 2013).
By exchangeability of the subjects, we can obtain the conditional prior distribution for \( \theta_i \) given \( \theta_{-i} \) as

\[
\theta_i | \theta_{-i} \sim \frac{\alpha}{\alpha + n - 1} P_0(\theta_i) + \sum_{h=1}^{k(-i)} \frac{n_h^{(-i)}}{\alpha + n - 1} \delta_{\theta_i^*, h}
\]

where \( \theta_i^* \) denotes the uniques values of \( \theta_{-i} \), \( n_h^{(-i)} \) represents counts of observations in cluster \( h \) excluding observation \( i \), and \( k(-i) \) denote the number of clusters excluding observation \( i \) (Gelman et al., 2013). Updating this full conditional prior with the data, we can obtain the conditional posterior distribution for \( \theta_i \). Alternatively, we can separately update the allocation of subjects to clusters and cluster-specific parameters, leading to the marginalised Gibbs sampler through the following alternate steps:

1. Update the allocation \( Z \) by sampling from the multinomial conditional posterior with

\[
p(z_i = h|\cdot) \propto \begin{cases} 
\frac{n_j^{(-i)} f(x_i | \theta_h^*)}{\alpha \int f(x_i | \theta) dP_0(\theta)} & \text{if } h = 1, \ldots, k(-i) + 1 \\
\frac{n_j^{(-i)} f(x_i | \theta_h^*)}{\alpha \int f(x_i | \theta) dP_0(\theta)} & \text{if } h = k(-i) + 1.
\end{cases}
\]

2. Update the unique values of \( \theta_h^* \) from

\[
\theta_h^* \sim P_0(\theta_h^*) \prod_{i \in \{z_i = h\}} f(x_i | \theta_h^*).
\]

Note that when \( z_i = h = k(-i) + 1 \), observation \( i \) is allocated to a singleton cluster (Gelman et al., 2013). This algorithm is only of practical use under conjugate models where base measure \( P_0 \) is conjugate with likelihood \( f(x_i | \theta_h^*) \) and the integral in second step can be evaluated (Fruhwirth-Schnatter et al., 2019). In such cases, the integral in step 1 can also be calculated analytically and the conditional posterior in step 2 also has the same parametric form as \( P_0 \) but with updated parameters (Gelman et al., 2013).
Appendix E

Additional materials on proposed DP-PMM

In this section, we provide additional discussion on proposed Bayesian mixture model for misclassified binary data. In Section E.1, we review how auxiliary variable can simplify posterior sampling in Bayesian probit model. In Section E.2, we provide more general discussion on the problem of label switching in Bayesian analysis of mixture models and common methods to handle label switching including the ECR algorithm used in our experiments as discussed in Section 4.1.4.

E.1 Auxiliary variable for Bayesian probit model

In Bayesian probit regression where \( p(T|\beta, X) = \Phi(X\beta) \), we can write

\[
\begin{align*}
  t_i & \sim \text{Bern}(\Phi(x_i\beta)), \\
  \beta & \sim \pi(\beta).
\end{align*}
\]
The posterior distribution for Bayesian binary probit model follows

\[
p(\beta|X, T) \propto p(\beta)p(T|\beta, X) = \pi(\beta) \prod_{i=1}^{n} \Phi(x_i\beta)^{t_i} (1 - \Phi(x_i\beta))^{1-t_i}
\]

To make sampling easier, \(n\) independent latent variables \(t_i^*\) are introduced where each \(t_i^*\) follows a normal distribution. The new hierarchical model can be written as

\[
t_i = \begin{cases} 
1 & \text{if } t_i^* > 0 \\
0 & \text{if } t_i^* \leq 0 
\end{cases}
\]

\[
t_i^* = x_i\beta + \epsilon_i \\
\epsilon_i \sim N(0, 1) \\
\beta \sim \pi(\beta)
\]

where now \(t_i\) is deterministic conditional on the sign of auxiliary variable \(t_i^*\). The joint posterior distribution of latent variable \(t^*\) and \(\beta\) has distribution:

\[
p(t^*, \beta|t, X) \propto p(\beta)p(t^*|\beta, X)p(T|t^*) = \pi(\beta) \prod_{i=1}^{n} p(t_i^*|\beta, x_i)p(t_i|t_i^*)
\]

where \(t_i^*|\beta, x_i \sim N(t_i^*|x_i\beta, 1)\) and \(t_i|t_i^* = \mathbb{1}(t_i = 1)\mathbb{1}(t_i^* > 0) + \mathbb{1}(t_i = 0)\mathbb{1}(t_i^* \leq 0)\).

The inference of interest here is the joint posterior of \(\beta, t^*\), which be sampled through Gibbs sampler. The full conditional

\[
p(\beta|t^*, t, X) = p(\beta|t^*, t, X) | \propto \pi(\beta) \prod_{i=1}^{n} N(t_i^*|x_i\beta, 1).
\]
With normal prior on $\beta$ where $\pi(\beta) = MVN(M_0, Q_0)$, we can write the posterior of $\beta$

$$\beta|t^*, X \sim MVN(M, V)$$

where

$$M = V \left( Q_0^{-1} M_0 + X^T t^* \right), \quad V = (Q_0^{-1} + X^T X)^{-1}.$$ 

For $t_i^*$, conditional on $\beta$ and $x_i$, it is just $N(x_i, \beta, 1)$ but given also $t_i^*$, it becomes truncated normal

$$t_i^*|\beta, t_i, x_i \sim \begin{cases} 
TN(x_i, \beta, 1) \in (0, \infty) & \text{if } t_i = 1, \\
TN(x_i, \beta, 1) \in (-\infty, 0] & \text{if } t_i = 0.
\end{cases}$$

### E.2 Label Switching

In Bayesian analysis of mixture models, additional caution need to be taken due to label switching problem. The *label switching problem* refers to the fact that in mixture models under symmetric prior, the resulting posterior will be invariant when labels are permuted (Papastamoulis and Iliopoulos, 2010). Since the posterior is proportional to the product of a symmetric likelihood with a symmetric prior, the marginal distribution will be all the same. The components are thus exchangeable as all the marginals on $\theta_j$ are identical. If the interest of inference is just the full mixture posterior, the label switching problem may not be of concern. However, the label switching problem cannot be ignored if cluster-specific parameters are of interest like the case of DP-PMM.

Algorithmically speaking, mixture models belong to the group of inverse problems where data provide information on the parameters indirectly. It is also ill-posed since small changes in the data may induce large changes in the results (Marin et al., 2005). Consider a sample of size $n$, there is a positive probability of $(1 - p_j)^n$ that the $j$th component is empty. In other words, there is positive probability that the sample brings no information about the parameters of one or more components.
To resolve the label switching problem, many methods have been proposed in literature. The majority of them deal with the label switching problem in the post-sampling phase by reordering the MCMC samples. Broadly speaking, these methods can be divided into three categories depending on the criteria used for reordering. In the first category, the parameters of the model $\theta$ and probabilities of cluster components $\pi$ are used for reordering. This can be simply ordering the parameters in descending or ascending order (artificial identifiability constraints) or permute samples so that the permuted parameters are as close to MAP estimates as possible. In the second category, reordering is based on the latent variable $z$ that determines cluster membership. In Papastamoulis and Iliopoulos (2010), MCMC samples are permuted on the assignments of latent clusters directly and in Rodriguez and Walker (2014), permutation are guided by pivoting on the location of the clusters, minimising a kNN-like distance among clusters centers and observations. In the third category, reordering is done in the probabilistic way where the unknown permutation $\pi(t)$ associated with the $(t)$th iteration of the MCMC sampler is viewed as having discrete density conditional on the data, $x$, and the full vector of parameters $\Theta = \{p, \theta\}$. Next, we describe the equivalence class representation algorithm and the data-based algorithm in more details and discuss why they are the only feasible algorithm to deal with label switching in DP-PMM later.

### E.2.1 Equivalence class representation algorithm

In our experiments, we use the *equivalence class representation* to deal with label switching which belongs to the category of methods that base reordering on latent variable $z$. The *equivalence class representation* (ECR) algorithm was originally proposed by Papastamoulis and Iliopoulos (2010) based on the idea that equivalent allocation vectors are mutually exclusive from the label switching solution. Two allocation vectors are equivalent if one can obtain the other by permuting its labels (Papastamoulis, 2015). The big idea behind ECR algorithm is to find the optimal permutation by pivoting on the cluster assignment.
Let $\mathbf{g} = (g_1, g_2, \ldots, g_n)$ denote the vector of true cluster allocation for each observation, the optimal permutation $p_t$ on simulated MCMC sample $(t)$ can be defined as the one that minimises the distance between permuted cluster assignments and true allocations. This objective can be written as maximising

$$\sum_{i=1}^{n} 1\left[g_i = p_t(z_i(t))\right].$$

The pivot used here is the true cluster allocations $\mathbf{g}$ which are not known. To estimate $\mathbf{g}$, Papastamoulis and Iliopoulos (2010) propose different estimation procedures that leads to different realisation of ECR algorithm, all implemented in R package `label.switching`.

The simplest way to estimate $\mathbf{g}$ is to use the MAP estimator $\mathbf{g}^\text{map} = \mathbf{z}^{(t^*)}$ where $t^*$ is the simulated MCMC sample where the log-posterior distribution is maximised, i.e.

$$t^* = \arg \max_{t \in \{1, \ldots, M\}} \log[p(\mathbf{\pi}^{(t)}, \mathbf{\theta}^{(t)}, \mathbf{z}^{(t)}|x)].$$

Once the iteration with maximum log-posterior distribution $t^*$ is found, the estimates true cluster allocation is set to the cluster assignment at that iteration, $\mathbf{g}^\text{map} = \mathbf{z}^{(t^*)}$. The optimal permutation $p_t$ is the one that minimises total number of mismatches to $\mathbf{z}^{(t^*)}$. Rather than using results from a single iteration to estimate true allocation $\mathbf{g}$, the cluster allocation of each observation $g_i$ can be estimated separately. For example, one choice is to find $\mathbf{g}$ such that each $\hat{g}_i$ is the mode of cluster assignment after permutation, or $\hat{g}_i = \text{mode} \{p_1(z_1^i), \ldots, p_M(z_M^i)\}$. Alternatively, $\hat{g}_i$ can be the cluster with highest average cluster probability. So $\hat{g}_i = j$ if $\frac{1}{M} \sum_{t=1}^{M} p_{t,i}(j)$ is the largest.

An alternative to using the MAP estimator as pivot in the equivalence class representation algorithm is to find a pivot iteratively. One can select a random pivot to start and then reset the pivot to the mode permutation in each iteration until no changes in the pivot can be
E.2.2 Data-based algorithm

Rodriguez and Walker (2014) propose a relabelling algorithm pivoting on the location of the clusters. The center and dispersion of the clusters are used as pivots and in each simulated MCMC sample the optimal permutation is defined to be the one that minimizes the loss function measuring the total distance from observations to the clusters. Rodriguez and Walker (2014) suggest a loss function that resembles the k-means clustering distance where

\[ L(X, m, s) = \sum_{j=1}^{K} n_{p(j)} \sum_{i: p(t_i) = j} \left( \frac{x_i - m_c}{s_c} \right)^2 \]

where \( m_1, \ldots, m_K \) denote cluster centers, \( s_1, \ldots, s_K \) the dispersions, and \( n_{p(j)} \) represents the number of observations in class \( j \) after the data are permuted by \( p_t \). The mean and dispersion of each cluster can be estimated by the expected posterior mean and variance where

\[ \hat{m}_c = \mathbb{E} \left( \frac{1}{n_j} \sum_{i: z_i = j} x_i \right), \quad \hat{s}_c^2 = \mathbb{E} \left( \frac{1}{n_j - 1} \sum_{i: z_i = j} (x_i - \bar{x}_j)^2 \right). \]

where \( n_j \) denotes the number of observations in cluster \( j \). This algorithm is deterministic and most computationally efficient.

E.2.3 Label switching algorithm consideration for DP-PMM

For DP-PMM, both the iterative ECR and data-based algorithm are good options to deal with label switching and are implemented in R Package \texttt{label.switching}. The other methods implemented in this package to deal with label switching are not suitable because they require additionally storing large matrices, is computationally expensive, or is not principled. \textit{AIC (artificial identifiability constraints)} relabels MCMC output by simply ordering a
specific parameter. It is not appropriate for DP-PMM since the choice of which parameters is arbitrary and the ordering is ad-hoc. PRA is a geometrically-based pivotal reordering algorithm. It requires MCMC samples and a pivot parameter vector. The pivot should be carefully constructed and correspond to a high-posterior density point. Stephen's algorithm aim to maximises the agreement of the permuted sample points on \( n \times K \) matrix of classification probabilities using the KL divergence. The input is \( n_{\text{mcmc}} \times n \times K \) matrix that gives allocation probabilities in each iteration which require large memory to store in all iterations. sjw is a probabilistic relabelling algorithm that treat MCMC output as observed data and permutation as unknown and use a EM-type algorithm to estimate the weights fro each permutation per MCMC data point. This algorithm is not suitable for DP-PMM since EM type algorithm are known to be sensitive to starting point and it also requires additional parameters like log-likelihood of mixture model. Consequently, the only practical methods to apply to DP-PMM for label switching are the data-based labelling and iterative ECR algorithm.

### E.3 \( t \)-distribution approximation for Bayesian logistic regression

One solution to the posterior computation of Bayesian logistic regression is to approximate the multivariate logistic with a multivariate \( t \) distribution. Using \( R \) to denote the correlation
matrix, the Bayesian LR model can be written as

\[ y_i = \mathbb{1}(z_{ij} > 0) \]
\[ z_{ij} \sim \beta^{(i)} x_{ij} + \log \left( \frac{F(t_{ij})}{1 - F(t_{ij})} \right) \]
\[ t_i \sim N(0, \phi_i^{-1} R) \]
\[ \phi_i \sim Gamma \left( \frac{v}{2}, \frac{v}{2} \right). \]

where \( F(\cdot) \) is the cdf of \( t_{ij} \). For approximation, the \( t \) link can be used where

\[ y_i = \mathbb{1}(z_{ij}^* > 0) \]
\[ z_{ij}^* \sim \beta^{(i)} x_{ij} + \sigma t_{ij} \]
\[ t_i \sim N(0, \phi_i^{-1} R) \]
\[ \phi_i \sim Gamma \left( \frac{v}{2}, \frac{v}{2} \right). \]

Posterior for the multivariate t-link model can be sampled where Gibbs steps are used to update \( \beta, t_i, \phi_i \) and Metropolis step is used to update \( R \) (O’Brien and Dunson, 2004)
Bibliography


208


