

ABSTRACT

LEE, JIYONG. Multilevel Dirichlet Process Linear Model (MDPLM) for Heterogeneous Data. (Under the direction of Dr. Simon M. Hsiang.)

Multilevel Dirichlet process linear model (MDPLM) is a Bayesian nonparametric regression model in *heterogeneous environments* where the exact data distribution is difficult to fixate. MDPLM adjusts its complexity on data assumption by using *Dirichlet process prior* and *layer extension*. Due to *Dirichlet process (DP) prior* over parameter space, MDPLM makes a class in heterogeneous population data to be discernible from one another in parameter. In addition, the layer extension enables MDPLM to fit complex data distribution by increasing model complexity. Therefore, MDPLM overcomes the difficulty of the traditional way of regression in finding a proper model to use.

Many regression models or techniques are not free from data assumption. Most of models have their own assumptions on data distribution and the violation of those assumptions results in degradation of model performance. Therefore, a modeler has to investigate data and to find the characteristics of data before (s)he determines a proper model to use. Thus, a traditional modeling process usually entails statistical analysis or data mining and, sometimes, requires the knowledge of experts in the field which data come from. However, since real-world data increase in complexity as well as in size and, also, contain a lot of uncertainties to resolve, it is very difficult or sometimes impossible to find a proper model by the traditional way. Moreover, since most regression models have a single set of assumptions, which is assumed to be true throughout data, they cannot deal with complex heterogeneous data. Considering such difficulties, we propose a generative probabilistic model which is flexible on data assumption.

Our research is to develop a prediction model and its methodology, which can be ap-

plicable to a complex data distribution even of unknown form. In order to remove the requirement of data assumption of a model, we construct a layered mixture model and adjust the complexity of a model by increasing the number of layers. Under the assumption that prediction error of a layer comes from low complexity of a model, we estimate the error in the next layer by increasing model complexity. While increasing complexity of a model, data are enhanced with information which can be drawn from covariates to reduce the variance of error. This data enhancement is called *covariate-extension* in our approach, and we define the information as a function of the difference in estimate between the mixture model of a layer and its overfit model. We call the function a *hidden random effect*, which represents the direction of model improvement because an overfit model has less bias in prediction and the differences of estimates between two models are estimates for errors of a layer.

An overfit model is constructed by disturbing a mixture model with designed noises during parameter sampling and its construction is related to the overfitting of the final layered mixture model. Avoiding loss of robustness to unseen data, we determine the degree of designed noises along with layers by solving a multi-stage decision problem.

In our approach, the model performance is measured by a cross-validation. Since it is difficult to obtain a well-stratified training-validation dataset pairs in a heterogeneous environment, we modify the sampling scheme by utilizing the degree of heterogeneity and use an entropy-based weighted sum of training-validation errors as our performance measure.

We name the integration of the layered mixture model and its methodology the *multilevel Dirichlet process linear model* (MDPLM). The proposed MDPLM differs from other regression models in the followings: (1) MDPLM does not require data assumption and, therefore, reduces modeling efforts such as data analysis or mining, (2) MDPLM adjusts model complexity according to complexity of data distribution by Dirichlet process prior and

layer-extension, (3) MDPLM, by integrating a model and performance measure, produces a robust prediction model avoiding overfitting, and (4) MDPLM resolves uncertainty of error measures in a cross-validation by using a different way of sampling training-validation data and an entropy-based performance measure.

We investigate the properties of MDPLM with simulated datasets in various situations: (1) a hidden random effect, (2) the distributions of different complexity, or (3) heteroscedasticity exist. Also, we make regression models with two real datasets using MDPLM to test its ability in heterogeneous environments.

© Copyright 2011 by Jiyong Lee

All Rights Reserved

Multilevel Dirichlet Process Linear Model (MDPLM) for Heterogeneous Data

by
Jiyong Lee

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Operations Research

Raleigh, North Carolina

2011

APPROVED BY:

Dr. Julie S. Ivy

Dr. Min Liu

Dr. Shu-Cherng Fang

Dr. Simon M. Hsiang
Chair of Advisory Committee

DEDICATION

Dedicated to

My parents, Jaesun Lee and Byunglim Jeon,

My parents-in-law, Jongdae Kim and Kyungae Park,

My wife, Soeyeon Kim, and

My son, Alexander Lee

BIOGRAPHY

Jiyong Lee was born on Nov. 7th, 1974 in Seoul, South Korea. He majored in electronics engineering and received his Bachelor of Engineering degree from Korea University in 2000. He came to the United States and earned his Master of Science degree in the Department of Computer Science at University of Southern California in 2004. He started his Ph.D. study in the Department of Computer Science at North Carolina State university in the fall of 2005 and joined the Operations Research program in the spring of 2009.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Simon M. Hsiang, for his great guidance and support throughout my Ph.D. study. I would also like to thank my wife, Soyeon Kim, because this work would not have been possible without her support and endurance. Finally, I thank my son, Alexander Lee, for giving me love and joy so that I can sustain myself in difficulties.

TABLE OF CONTENTS

List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	4
1.3 Overview of Research	5
1.3.1 Research Objective	5
1.3.2 Assumptions of Research	6
1.3.3 Methods	6
1.4 Contributions of Research	8
1.5 Overview of Chapters	10
Chapter 2 Literature Review	12
2.1 Heterogeneity and a Bayesian modeling	13
2.1.1 Heterogeneous Population	13
2.1.2 The Number of Classes and a Generative Nonparametric Modeling	15
2.2 Regression Models for Heterogeneous Population Data	15
2.2.1 Mixed Models	15
2.2.2 Multilevel Mixed Models	17
2.2.3 Dirichlet Process Prior Regression Models	17
2.3 Heterogeneity and Parameter Estimation	19
2.3.1 Parameter Estimation	19
2.3.2 Bayesian Parameter Estimation	20
2.4 Adaptive Rejection Sampling	22
2.5 Dirichlet Process Prior Models	26
2.5.1 Dirichlet Process	26
2.5.2 Dirichlet Process Prior Mixture Models	28
2.5.3 Dirichlet Process Prior Regression Models	37
Chapter 3 Multilevel Dirichlet Process Linear Models	45
3.1 Description	46
3.2 Structure of Multilevel Dirichlet Process Linear Model	48
3.3 Covariate Extension of Layers	49
3.4 Overfit Models of Layers	52
3.5 The Layer of Prediction	54
3.5.1 Performance Measure	54

3.5.2	Optimal Path to the Layer of Prediction	55
3.5.3	Multi-stage Decision Problem for the Layer of Prediction	57
3.5.4	Generating training-validation dataset pairs	61
3.5.5	An Alternative Way for the Layer of Prediction	62
Chapter 4	Properties of MDPLM	71
4.1	Hidden Random effects	71
4.2	Different Relationships	80
4.3	Heteroscedasticity	84
Chapter 5	MDPLM Applications	90
5.1	MDPLM for Preliminary Engineering Cost	90
5.1.1	Background	90
5.1.2	Data Description	91
5.1.3	Procedure and Results	94
5.2	MDPLM for Capsular Penetration in Prostate Cancer	96
5.2.1	Background	96
5.2.2	Data Description	97
5.2.3	Procedure and Results	99
Chapter 6	Summary and Conclusions	104
6.1	Summary of MDPLM Procedure	104
6.2	Discussion	108
6.2.1	Heterogeneity and Model Assumptions in MDPLM	108
6.2.2	Cross-validation and Degree of Heterogeneity in MDPLM	112
6.3	Conclusions	118
6.4	Future Research	120
References	126

LIST OF TABLES

Table 1.4.1	Comparison of models	10
Table 2.1.1	Class and parameter inference	14
Table 2.5.1	The prior distributions and corresponding posteriors for parameters in Dirichlet process linear models	44
Table 4.1.1	Comparisons of error measures between DPLM and MDPLM for data of hidden random random effect	79
Table 4.2.1	Comparisons of error measures between DPLM and MDPLM for data of different relationships	82
Table 4.3.1	Comparisons of error measures between DPLM and MDPLM for data of heteroscedasticity	88
Table 5.1.1	Data sources for bridge projects	92
Table 5.1.2	The list of variables for bridge model	93
Table 5.1.3	The designed noise rates of bridge projection	94
Table 5.1.4	Comparison of error measurements for the bridge project model between layer 0 and 4	95
Table 5.1.5	The error measurements of the final bridge MDPLM for preliminary engineering cost ratio estimation	95
Table 5.2.1	The list of variables for the prostate capsule penetration model	98
Table 5.2.2	Confusion matrix	99
Table 5.2.3	The designed noise rates of the prostatic capsular penetration MDPLM	100
Table 5.2.4	Confusion matrices of MDPLMs with the different layers of prediction	101
Table 5.2.5	Logit models for prostatic capsular penetration	103

LIST OF FIGURES

Figure 2.5.1	The conditional dependency of parameters in Gaussian mixture models	33
Figure 2.5.2	The conditional dependency of parameters in DPLMs	42
Figure 3.2.1	The schematic of MDPLM	49
Figure 3.5.1	The plots of designed noise rate vs error measures at layer 1 for data of Eq. 4.1.1	66
Figure 3.5.2	The plots of $E(cMSE)$ sorted in increasing order	67
Figure 3.5.3	The MDPLM system	67
Figure 3.5.4	Transition of error measures by layers (2)	70
Figure 4.1.1	The plot for (x_1, x_2, y) in Eq. 4.1.1	73
Figure 4.1.2	The plot for the mean parameter (μ) of covariate $X = (x_1, x_2, z)$ over sampling in DPLM	74
Figure 4.1.3	The distribution of z over sampling in DPLM of covariate $X = (x_1, x_2, z)$	74
Figure 4.1.4	The overlap of covariate regions	76
Figure 4.1.5	The number of mixture components (K) over iterations in the model with covariate $X = (x_1, x_2)$	77
Figure 4.1.6	Prediction comparison of DPLM between $X = (x_1, x_2)$ and $X = (x_1, x_2, z)$	77
Figure 4.1.7	Performance measures by layers	78
Figure 4.1.8	Comparison of estimates between layer 0 and 1	78
Figure 4.2.1	The plot for (x, y) of Eq. 4.2.1	81
Figure 4.2.2	The covariate region of DPLM for Eq. 4.2.1 at iteration 3000	82
Figure 4.2.3	Performance measures by layers	83
Figure 4.2.4	Comparison of response estimate between DPLM and MDPLM	83
Figure 4.3.1	The plot for $y = \log(1 + x + e(x))$	86
Figure 4.3.2	The performance measures of MDPLM over layers	87
Figure 4.3.3	The plot for the number of components over parameter sampling	87
Figure 4.3.4	Comparison of response estimates between DPLM and MDPLM	88
Figure 4.3.5	The plot for $\mathcal{N}(\mu, s^{-2})$ over sampling	89
Figure 4.3.6	Sampled regression lines	89
Figure 5.1.1	Determining the layer of prediction for the bridge construction model	95
Figure 5.1.2	The layer-wise predictive results of the bridge MDPLM trained with 505 construct projects	96
Figure 5.2.1	Determining the layer of prediction for the prostate cancer model	101
Figure 6.2.1	Relation of data, heterogeneity, and model complexity	109

Figure 6.2.2	A general approach for modeling.	110
Figure 6.2.3	A generative probabilistic approach for modeling.	112
Figure 6.2.4	MDPLM as a self-adjusting probabilistic approach for modeling.	113
Figure 6.2.5	<i>cMSE</i> error measure	117

CHAPTER 1

Introduction

Multilevel Dirichlet process linear model (MDPLM) is a Bayesian nonparametric regression model in *heterogeneous environments* where the exact data distribution is difficult to fixate. MDPLM adjusts its complexity on data assumption by using *Dirichlet process prior* and *layer extension*. Due to *Dirichlet process (DP) prior* over parameter space, MDPLM makes a class in heterogeneous population data to be discernible from one another in parameter. In addition, the layer extension enables MDPLM to fit a complex data distribution by increasing model complexity. Therefore, MDPLM overcomes the difficulty of the traditional way of regression in finding a proper model to use.

1.1 Motivation

Most regression models or techniques are not free from data assumptions, for examples:

1. The error is a random variable with a mean of zero conditional on the explanatory variables.
2. The errors are uncorrelated, that is, the variance-covariance matrix of the errors is diagonal and each non-zero element is the variance of the error.
3. The variance of the error is constant across observations (homoscedasticity).
4. The independent variables are measured with no error.
5. The predictors are linearly independent, i.e. it is not possible to express any predictor as a linear combination of the others.
6. The sample is representative of the population for the inference prediction.

If the assumptions of a model fail to follow real-data distribution, the performance of models will be degraded. However, the method can still be used even though the assumptions are not true. Variation from the assumptions can sometimes be used as a measure of how far the model is from being useful. Many of these assumptions may be relaxed in more advanced treatments. Therefore, a modeler has to investigate data and to find characteristics of data before (s)he determines a proper model to use. Documentations of analyses usually include analyses of tests on the sample data and methodology for the fit and usefulness of the model. Hence, the traditional modeling process usually entails statistical analysis and/or data mining and, sometimes, requires the knowledge of experts in the field which data come from.

However, since real-world data increase in complexity as well as in size and contain a lot of uncertainties to resolve about distribution, it is very difficult or sometimes impossible to find a proper model for regression in a traditional way. Moreover, since most regression models have a single set of assumptions, which is assumed to be true throughout the whole data, they cannot deal with complex heterogeneous data of more than one distinct datasets.

Considering such difficulties, it is necessary to develop a model, less constrained to data assumptions, for heterogeneous data.

Currently, there are several regression models for heterogeneous data: *linear mixed model* (LMM), *generalized linear mixed model* (GLMM), *multilevel linear mixed model* (multilevel-LMM), *Dirichlet process linear model* (DPLM), and *Dirichlet process generalized linear model* (DPGLM). However, they are not sufficient to handle a real-world complex heterogeneous dataset which is filled with uncertainties about data distribution.

LMM has strict data assumption; there is a known number of classes, all response functions are linear, random error of a class is *i.i.d.* As an extension of LMM, GLMM is more flexible by replacing the linear terms of LMM with *generalized linear models* (GLM). Thus, GLMM can handle nonlinearity and *heteroscedasticity* (change of variance). However, in order to use GLMM, we still need the *degree of heterogeneity* (the number of classes) and the forms of nonlinear and error functions. Thus, if data are too complex to draw any information, LMM and GLMM will not be used for regression modeling. Different from other mixed models, multi-level LMM can model variance relationships between classes; however, it requires the knowledge of how data are structured. Considering uncertainties of heterogeneous data about the degree of heterogeneity and the forms of response and error functions, any mixed model is not appropriate for real-world complex data. *Dirichlet process* (DP) prior models, DPLM and DPGLM, are less constrained than mixed models in that the degree of heterogeneity is not required for modeling. However, since response functions and error-distributions are assumed to be known in uniform forms, DP prior models cannot handle the cases that response or error-distribution consists of functions of different forms.

In this dissertation, we propose a model and its methodology which are not dependent on data assumption. The proposed model overcomes the difficulty of the traditional way of regression in finding a proper model to use and makes unnecessary data analysis or experts'

knowledge before modeling.

1.2 Problem Statement

Difficulties of modeling in a heterogeneous environment come from uncertainties about a population distribution:

1. Covariate-response function
2. Forms of random errors
3. Random effects
4. Interactions within covariates
5. Hidden variables
6. Hidden random effects
7. Degree of heterogeneity (the number of classes).

A regression model has its own data assumptions corresponding to the uncertain factors of data distribution listed above. For instance of a linear regression, the response is a linear function of covariates, the distribution of random error is an *independent and identically distributed (i.i.d.)* normal centered at the mean of responses given covariate values, there is no random effect, interactions and hidden variables, and the degree of heterogeneity is one. Therefore, how well a regression model will predict depends on how well data assumptions of the model follow real-data distribution.

However, making decisions on such factors is not easy in a heterogeneous environment because those factors are interwoven and having an effect on the performance of a model in a complicated way. Using a complex relationship will reduce the degree of heterogeneity and achieve a good performance value for training (or observed) data, but may result in

bad performance to unseen data due to possible overfitting. In a case that data do not directly reveal the relationship between covariate and response, finding a hidden variable or transforming data may improve the performance of a model. However, such increase in model complexity will also cause overfitting when the relationship between modified covariate (by enhancing or transforming) and response is valid only in the training data. When covariate-response relationship is fixed, the increase on the degree of heterogeneity will increase possibility of overfitting. On the other hand, when covariate-response relationship is free to choose, the increase on the the degree of heterogeneity will reduce the complexity of relationship function and, hence, may avoid overfitting. Therefore, there should be a trade-off among data assumptions of a model, and a model, obtained by locking up some of data assumptions, may experience underfitting or overfitting, depending on how good data assumptions represent real-data distribution.

1.3 Overview of Research

This section will discuss our research briefly on developing the proposed model, called *multilevel Dirichlet process linear model*.

1.3.1 Research Objective

The objective of this dissertation is to develop a Bayesian nonparametric regression model and its methodology, which are applicable to heterogeneous environments where it is difficult to find the exact data distribution. By integrating the modeling procedure and the performance measure into one framework, we propose to make a layered model which adjusts its complexity according to data distribution. In addition, different from other models which require data assumptions, the proposed model does not have any data assumption.

Hence, the model does not require data analysis and experts' knowledge, so extends its applicability to very complex data which are full of uncertainties.

This research has both designing and theoretical aspects. The designing aspect involves the creation of a new framework in a Bayesian nonparametric environment to find the direction of model-improvement by utilizing an overfit model and to generate a hidden random effect. The theoretical aspects involves the formulation of a multi-stage decision problem for the layer of prediction and the efficient way of solving the problem in the proposed framework. Additionally, this research introduces another view of performance estimation in cross-validation, and the stratified sampling on heterogeneous data , which are related to theoretical and designing aspects, respectively.

1.3.2 Assumptions of Research

Throughout the dissertation, we have the following assumptions about data.

1. *A heterogeneous dataset is a collection of datasets collected from different sources.*
2. *Data from the same source share the same statistical characteristics.*

The assumptions say that a heterogeneous dataset is the mix of characteristics from different data distributions. Rather than assuming that a data distribution can be parameterized as in other descriptive methods, we use a nonparametric way to represent a data distribution via a mixture of data assumptions, which makes it possible to represent a complex distribution with less complex ones.

1.3.3 Methods

This research involves the following topics: *Bayesian nonparametric regression, Monte Carlo Markov Chain (MCMC) sampling, non-MCMC sampling, Markov Chain (MC) integral, model*

selection, cross-validation (CV), multi-stage optimization, stochastic resonance theory, bias-variance tradeoff. Also, the research draws a concept from *control theory* to generate an overfit model and to solve a multi-stage optimization problem. *Data-mining* is also involved in that MDPLM extracts a hidden random effect from a model and its overfit model. Thus, this research integrates those topics and studies into a model and its methodology.

We will develop *multilevel Dirichlet process linear model* (MDPLM) in Chapter 3, and the followings are methods in developing steps. Each method is followed by its referential sections and brief explanations.

1. *A multilevel Bayesian nonparametric model is formulated.* - Section 3.2

The proposed model MDPLM consists of the layers of *Dirichlet process linear models* (DPLMs). MDPLM is extendible in two directions: parameter and data-assumption. By applying a *Dirichlet process prior* over parameter space, a layer constructs a mixture of linear models and data distribution determines the number of mixture components. Since each layer has the same assumptions as DPLM, the violation of data assumption in a layer is remedied by extending layers.

2. *MDPLM utilizes overfit models to estimate the error of a layer.* - Section 3.4

A layer of MDPLM has two kinds of DPLM: a generic DPLM and an overfit DPLM, denoted by DPLM and oDPLM, respectively. An oDPLM has less bias but more variance in prediction than its counterpart DPLM. The difference of response estimates between DPLM and oDPLM is an estimate for error of the layer.

3. *Based on the estimated error of a layer, a hidden random effect is generated to represent the fit of the previous layer.* - Section 3.3

Data are enhanced to contain the information about how well a current DPLM fits target values by estimating error with aid of an oDPLM.

4. *The variance of error is reduced by adding a layer.* - Section 3.3

The data, enhanced with a hidden random effect, help reduce the variance of error in the regression of the next layer.

5. *M-fold cross-validation is used with a different way of constructing training-validation dataset pairs and evaluating a model for a less biased performance measure.* - Section 3.5.1 and Section 3.5.4

We use the modeling result of DPLM to retrieve the degree of heterogeneity and construct training-validation dataset pairs in order to prevent a training dataset from not having any information of a class. We also evaluate a layered model by our performance measure, the entropy-based weighted sum of two mean square errors from training and validation data, to give more credits to a measure of certainty.

6. *By solving a decision problem about an optimal path of designed-noise rates and the layer of prediction, a resulting model will balance bias-variance in prediction.* - Section 3.5.3 and Section 3.5.4

The overfitting level of oDPLM is controlled by designed-noise rate, and models are different in performance depending on the overfitting levels of oDPLMs in layers. MDPLM solves a multi-stage decision problem to find the sequence of overfitting levels to provide a robust model in the view of bias-variance tradeoff.

1.4 Contributions of Research

The proposed MDPLM does not require data assumption. MDPLM is the integration of a generative probabilistic model and a performance estimation procedure to model complicated heterogeneous data. Existing descriptive modeling approaches require data assumptions

which are obtained by data analysis or mining, but it is difficult or sometimes impossible to find the exact data distribution in heterogeneous environments. We overcome the difficulty of the traditional way of regression modeling by applying Dirichlet process prior and layer extension. MDPLM utilizes overfitting models to find the direction of model improvement and adjusts complexity of a model to fit data distribution. Therefore, MDPLM does not force any data assumption which take a risk from discrepancy between data assumptions and real data distribution.

MDPLM avoids overfitting and produces a robust regression model. As a non-descriptive method, *multilayer perception* (MLP) has difficulties in finding a proper design and the number of iterations to avoid under- or over-fitting. We avoid such a problem by integrating a model and a performance estimation procedure. The increase of model complexity in MDPLM is based on bias-variance tradeoff. MDPLM solves a multi-stage decision problem for a performance measure and determines the layer of prediction to prevent the complexity of model from diverging.

MDPLM resolves uncertainty of error measures. We modified a cross-validation (CV) to provide less biased error measurements. A reason for biased error measurements of CV in heterogeneous data is unbalanced information in training-validation datasets. We employ the degree of heterogeneity to avoid the worst case of sampling that a training dataset does not contain any information from a class. In addition, we use an different performance measure which is a weighted sum of training-validation errors by entropy. Therefore, MDPLM estimates the performance of a model with less bias and the robustness of the resulting model increases.

In summary, MDPLM and its procedure are applicable to a complex data distribution even of unknown form, and the resulting model will be robust in prediction to unseen data. The comparison with other descriptive models are presented in Table 1.4.

Table 1.4.1: Comparison of models

Model	Response functions	Error distributions	Data assumption	Deg. of heterogeneity	Lev. of modeling efforts
LM	linear	homoscedasticity	yes	fixed	high
GLM	nonlinear	hetroscedasticiy	yes	fixed	high
Multilevel LM	linear	homoscedasticity	yes	fixed	high
DPLM	mix. of linear	homoscedasticity	yes	variable	middle
DPGLM	mix. of nonlinear	heteroscedasity	yes	variable	middle
MDPLM	dependent	heteroscedasticity	no	variable	low

1.5 Overview of Chapters

Chapter 2: Literature Review In this chapter of the dissertation, the literature on heterogeneity and regression models is discussed. In Section 2.1, we discuss the meaning of heterogeneity and the advantage of a Bayesian nonparametric modeling in heterogeneous environments. In Section 2.2, we discuss regression models and their characteristics in heterogeneous environments. In the following Section 2.3 and Section 2.4, we discuss parameter estimation in a descriptive and a Bayesian model, and a non-MCMC sampling method for non-conjugate prior distributions. Dirichlet process prior models are presented in Section 2.5 with detailed derivation.

Chapter 3: Multilevel Dirichlet Process Linear Models We develop the *multilevel Dirichlet process linear model* (MDPLM) in the chapter. The structure of MDPLM is presented in Section 3.2, and how the variance of prediction error reduces is explained in Section 3.3. In Section 3.4, how an overfit model is constructed in MDPLM is presented. In Section 3.5, we discuss how to construct and solve a multi-stage decision problem for the layer of prediction.

Chapter 4: Properties of MDPLM Properties of MDPLM are tested with three simulated datasets. We compare MDPLM with DPLM in data distributions of several properties: hidden

random effect (hidden interaction), diversity of function complexity, and heteroscedasticity.

Chapter 5: MDPLM Applications In the chapter we use MDPLM to construct regression models for real-world complex datasets to test its ability in heterogeneous environments. In Section 5.1, we illustrate how to model data of unknown distribution using MDPLM. In Section 5.2, the application of MDPLM as a classifier will be shown comparing with logit models.

Chapter 6: Summary and Conclusions We summarize and conclude the dissertation. In Section 6.1 the MDPLM procedure is summarized in a step-by-step manner, and in Section 6.2 MDPLM is discussed in two points of view such as model assumption and bias-variance tradeoff. In Section 6.3 the contributions of MDPLM are discussed in detail. Finally, we discuss possible future studies with elaboration in Section 6.4.

CHAPTER 2

Literature Review

This chapter provides the overview of regression models on heterogeneous data. In Section 2.1, we discuss the meaning of heterogeneity and the advantage of a Bayesian non-parametric modeling in heterogeneous environments. In Section 2.2, we discuss regression models and their characteristics in heterogeneous environments. In the following Section 2.3 and Section 2.4, we discuss parameter estimation in a descriptive and a Bayesian model, and a non-MCMC sampling method for non-conjugate prior distributions. Dirichlet process prior models are presented in Section 2.5 with detailed derivation.

2.1 Heterogeneity and a Bayesian modeling

2.1.1 Heterogeneous Population

Heterogeneity implies the lack of consistency in the concept of interest from different populations. With respect to data, two possible interpretations exist. On the one hand, data are heterogeneous in type and scale. A dataset collected from different sources may have different data types and scales. For instance, the height of a group is measured in feet, but the height of another group is measured in two levels such as ‘tall’ and ‘short’.

On the other hand of interpretation, heterogeneity is based on statistical difference. When a group of populations is modeled by distributions within a single specified parametric family (assuming a parameter has a normal distribution), their homogeneity (or heterogeneity) can be tested for possessing the same parameter by a statistical test such as *F-test*. When observations’ membership (to class) is not known but their distributions are of known forms, a partition of data and parameters of classes can be obtained by minimizing a loss measure or maximizing a likelihood measure for a given number of classes (Table 2.1.1). If a population has more than one distinct distribution, it is called a *heterogeneous population*, a group that shares the same distribution a *class*, and the number of classes *the degree of heterogeneity*.

For instance, suppose that $\mathbf{O} = \{o_1, \dots, o_n\}$ is a statistically exchangeable set and X_k has a Gaussian distribution with a distinct location parameter μ_k and a common scale parameter σ^2 , that is, $X_k \sim \mathcal{N}(\mu_k, \sigma^2)$, for $k = 1, \dots, K$. The heterogeneous population can be decomposed by maximizing the likelihood as follows.

$$\max_{c, \phi} \prod_{k=1}^K \prod_{\{o_i \in \mathbf{O} | c_i = k\}} \varphi(o_i | \mu_k, \sigma^2) \quad (2.1.2)$$

where $\phi = \{\mu_1, \dots, \mu_K\}$, and $\varphi(\cdot | a, b)$ denotes a probabilistic density function (pdf) of Gaussian

Table 2.1.1: Class and parameter inference

Assuming a statistically exchangeable data $\mathbf{O} = \{o_1, \dots, o_n\}$ to be a sample from heterogeneous population of K classes, the problem to minimize a loss measure $LOSS$ can be described by

$$\begin{aligned} & \min_{c, \phi} LOSS(c, X) & (2.1.1) \\ \text{s.t. } & X_k \sim F(\phi_k), \quad k = 1, \dots, K \end{aligned}$$

where

K : the number of classes (known),

$F(\phi_k)$: the distribution of X_k parameterized by $\phi_k, k = 1, \dots, K$

$\phi = \{\phi_1, \dots, \phi_K\}$: the set of distinct class parameters $\phi_k, k = 1, \dots, K$

X_k : a random variable for class $k, k = 1, \dots, K$

$X = (X_1, \dots, X_K)$: the vector of random variables of K classes,

c_i : the class index of o_i to denote that o_i is a sample from X_{c_i} , and

$c = (c_1, \dots, c_n)$: the vector of class indices for observations in \mathbf{O} .

distribution with location a and scale b .

There are two useful data analysis techniques for heterogeneous population data: classification and clustering. Given a partition of data, \mathbf{D} and c , finding a function (or relation) or the set of distinct parameters (ϕ_k) for a given form ($F(\cdot)$) to satisfy the relationship from \mathbf{O} to c is called classification. On the other hand, given the relation and data, $F(\cdot)$ and \mathbf{O} , finding class identification (c) and parameters (ϕ_k) is called clustering. Thus, when a heterogeneous population data does not contain class identification, clustering is useful to reveal it.

In the rest of this dissertation, we will confine the meaning of *heterogeneous* to statistical difference and use the term *heterogeneous data* for heterogeneous population data. In addition, we assume that the class identification is not directly observable in heterogeneous data and that heterogeneous data may contain more than one different form of distributions.

2.1.2 The Number of Classes and a Generative Nonparametric Modeling

In heterogeneous data, the degree of heterogeneity (or the number of classes) is usually an unknown factor to be estimated. By using clustering algorithms, this uncertainty may be resolved, but consistency of assumptions should be maintained between a model and a clustering algorithm because the degree of heterogeneity depends on other data assumptions.

Even though a model and a clustering algorithm share the same assumptions and the degree of heterogeneity is obtained, it is too strict to choose the most likely case as clustering algorithms do. Because we are ignorant of real-data distribution, the set of data assumptions of a model is a rough representative of real-data distribution. If we disregard other cases with high probabilities to occur except the most likely one, data distribution represented by a model will depart further from real-data distribution. Such a problem can be mitigated by applying a nonparametric modeling.

Different from an approach with a clustering algorithm, a Bayesian nonparametric approach allows variation on the degree of heterogeneity and provides a reasonable way to express a probabilistic relationship between the degree of heterogeneity and other parameters. Even when data assumptions of a model does not well represent real-data distribution, the variation of parameters can reduce the effect of a wrong data assumption.

2.2 Regression Models for Heterogeneous Population Data

2.2.1 Mixed Models

A *mixed model* is a statistical model which utilizes random effects. Apart from fixed models assuming that observations are independent and identically distributed, a mixed model assumes that there exist two sources of variations *within* and *between* classes. A *linear mixed*

model (LMM) is the simplest form of mixed models where the response function is linear in terms of both fixed and random effects as follows.

$$\begin{aligned}
 \mathbf{y}|\mathbf{u} &\sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}, \mathbf{e}) \\
 \mathbf{u} &\sim \mathcal{N}(\mathbf{0}, \mathbf{G}) \\
 \mathbf{e} &\sim \mathcal{N}(\mathbf{0}, \mathbf{R})
 \end{aligned}
 \tag{2.2.1}$$

where \mathbf{G} and \mathbf{R} are covariance matrices of fixed and random effects; \mathbf{X} and \mathbf{Z} are known design matrices for fixed and random effects; \mathbf{y} and \mathbf{e} are the response vector and its random error vector; $\boldsymbol{\beta}$ and \mathbf{u} are the regression coefficient vectors for fixed and random effects.

Assuming that \mathbf{u} and \mathbf{e} are uncorrelated, Eq. 2.2.1 can be written by

$$\mathbf{y}|\mathbf{u} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}\mathbf{G}\mathbf{Z}^T + \mathbf{R}).
 \tag{2.2.2}$$

When the variance components \mathbf{R} and \mathbf{G} are known, the coefficient parameters $\boldsymbol{\beta}$ and \mathbf{u} are estimated and predicted, respectively, by the *best linear unbiased estimator* (BLUE) or the *best linear unbiased predictor* (BLUP) [1] (conventionally, the term *estimate* is used for fixed effect, and *predict* is used for random effect). On the other hand, the variance components can also be estimated and predicted by the *maximum likelihood estimator* (MLE) and the *restricted maximum likelihood* (REML) estimator. However, since MLE tends to yield biased results because it assumes that all fixed effects are known without error, which is impossible in practice, the REML estimator, which eliminates the bias from fixed effects by data transformation, is preferred. In computation, since those parameters are not separable, especially in multi-dimensional space, $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, \mathbf{R} and \mathbf{G} are computed iteratively.

LMM has strict data assumptions: the relationship between response and covariate (the

fixed or random effects) is linear, the variance component is independent of fixed or random effect covariates, and a random effect has a normal distribution. The violation of any of those assumptions results in poor modeling results. As an extension of a linear mixed model, a *generalized linear mixed model* (GLMM) was developed by replacing the linear term with a *generalized linear model* (GLM) [2] and it became possible to model heterogeneous data of more complications. In GLMM, the inverse of *link function* maps the linear predictor of fixed and random effects to the mean of response nonlinearly. Thus, the nonlinear relationship is possible between covariate to response. Also, by including the variance component to the linear predictor, heteroscedasticity such as over-dispersion can be dealt with. Additionally, the assumption about a random effect is relaxed for a random effect to have more kinds of distribution. Such relationships among distribution, link function, and variance function for exponential family are well formulated in [3].

2.2.2 Multilevel Mixed Models

In general, many multivariate datasets are collected by complicated sampling designs, and it is common to use cluster sampling if the target population has a hierarchical nested structure. Goldstein [4, 5, 6] proposed the multilevel mixed model for modeling such a hierarchical structured dataset. While most mixed models else than multilevel mixed models neglect the dependency between random errors because of computational complexity, the multilevel mixed model incorporates the dependency by cross-level variance components.

2.2.3 Dirichlet Process Prior Regression Models

The mixed or multilevel mixed models assume that the structures of datasets are known, that is, the explanatory variables (or covariate) for random effects are fixed and the number

of classes is known. However, as mentioned in Section 2.1.2, the degree of classes cannot be known without data analyses or mining on heterogeneous data. Such a difficulty can be solved in a nonparametric way by setting the number of components as a hidden variable with a reasonable prior. As a well-known problem of nonparametric methods is lack of analytical forms in most cases because latent variables should be integrated out. However, non-analytical integrals can be approximated reasonably in many ways such as numerical integrations (including Laplace's approximation) and Markov Chain Monte Carlo (MCMC) integration.

As a nonparametric way of regression modeling, the Dirichlet process prior regression model [7, 8, 9] is a Bayesian approach which can deal with uncertainty about the degree of heterogeneity. Different from usual mixed models, Dirichlet process prior regression models are flexible in the degree of heterogeneity because the number of mixture components is determined by observed data distribution, which is the reason that such models are called infinite mixture models even though the number of mixture components is limited to the number of observations. The other benefit of Dirichlet process prior regression models, it is unnecessary to decide random effects. Whether an explanatory variable is related to random effects is decided by the posterior distribution of response, which minimizes the mean square error of response. Although the covariate-response relationship needs to be decided before modeling, the mixture property enables a complex relationship to be fitted with relationships of low complexity. Therefore, even in case that the data assumption of a model violates real data distribution mildly, it is possible to fit the data well. When variance component is not constant (heteroscedasticity), mixed models have to be tuned by selecting a proper nonlinear relationship function or modifying variance term. However, Dirichlet process prior models accommodate mild heteroscedasticity natively by increasing the number of mixture components.

2.3 Heterogeneity and Parameter Estimation

2.3.1 Parameter Estimation

Parameter estimation is to find the set of parameters to minimize (or maximize) a loss (or likelihood) measure throughout any possible assignment of observations to classes. Therefore, parameter estimation plays an important role in finding heterogeneity by partitioning data into groups of distinct parameters.

Given the number of classes, denoted by K , the maximum likelihood estimation for the parameter vector ϕ for a heterogeneous dataset \mathbf{X} as follows.

$$\begin{aligned}
 \max_{c, \phi} \mathcal{L} &= p(\mathbf{X}|\phi) & (2.3.1) \\
 &= \prod_{j=1}^K \prod_{i=1}^N p(x_i|\phi_j) p(c_i = j) \\
 &= \prod_{j=1}^K \prod_{i=1}^N p(x_i|\phi_j) I(c_i = j) \\
 &= \prod_{j=1}^K \prod_{\{x_i \in \mathbf{X} | c_i = j, i=1, \dots, N\}} p(x_i|\phi_j)
 \end{aligned}$$

and equivalently

$$\max_{c, \phi} \log(\mathcal{L}) = \sum_{j=1}^K \sum_{\{x_i \in \mathbf{X} | c_i = j, i=1, \dots, N\}} \log p(x_i|\phi_j). \quad (2.3.2)$$

where c_i denotes the class of observation x_i , $c_i \in \{1, \dots, K\}$ for $i = 1, \dots, N$; $c = (c_1, \dots, c_N)$ is the vector for class identifications of observations; $I(cond)$ is the indicator function, which is 1 when $cond$ is true or 0 otherwise.

Therefore, the maximum likelihood estimator $\hat{\phi}_{MLE,j}$ for parameter of class j can be obtained

by solving the following equations.

$$\frac{\partial \log(\mathcal{L})}{\partial \phi_j} = 0, \quad j = 1, \dots, K. \quad (2.3.3)$$

When K is unknown, the maximum likelihood estimator does not work any more because the log-likelihood in Eq. 2.3.4 will be maximized when $K = N$. Hence, parameter estimation needs a penalty term to avoid the increase of K . However, it is not well defined because K depends on the distributions (or function) $p(\cdot|\phi_j)$.

$$\max_{K,c,\phi} \log(\mathcal{L}) = \sum_{j=1}^K \sum_{\{x_i \in \mathbf{X} | c_i=j, i=1, \dots, N\}} \log p(x_i|\phi_j) \quad , \text{ for } K = 1, 2, 3, \dots, N. \quad (2.3.4)$$

2.3.2 Bayesian Parameter Estimation

Suppose $\hat{\phi}(X)$ is an estimator for an unknown univariate parameter Φ given a random variable X , and Φ has a prior distribution $G(\phi)$. The expected squared deviation of $\hat{\phi}(X)$ from Φ is given by

$$\begin{aligned} E \left[\{\hat{\phi}(X) - \Phi\}^2 \right] &= E \left[E \left[\{\hat{\phi}(X) - \Phi\}^2 | \Phi \right] \right] \\ &= \int \sum_x p(x|\phi) \{\hat{\phi}(x) - \phi\}^2 dG(\phi) \\ &= \sum_x \int p(x|\phi) \{\hat{\phi}(x) - \phi\}^2 dG(\phi). \end{aligned} \quad (2.3.5)$$

The integral part of Eq. 2.3.5 can be also rewritten by

$$\begin{aligned}
\int p \cdot (\hat{\phi} - \phi)^2 dG &= \int p \cdot (\hat{\phi}^2 - 2\hat{\phi}\phi + \phi^2) dG \\
&= \hat{\phi}^2 \int p dG - 2\hat{\phi} \int p\phi dG + \int p\phi^2 dG \\
&= \int p dG \left(\hat{\phi} - \frac{\int p\phi dG}{\int p dG} \right)^2 + \left[\int p\phi^2 dG - \frac{(\int p\phi dG)^2}{\int p dG} \right]. \quad (2.3.6)
\end{aligned}$$

Therefore, Eq. 2.3.6 is minimized by

$$\hat{\phi} = \frac{\int p(x|\phi)\phi dG(\phi)}{\int p(x|\phi) dG(\phi)} \quad (2.3.7)$$

and Eq. 2.3.5 is also minimized by the estimator.

Eq. 2.3.7 is called the Bayes estimator and is nothing but the expected value of the posterior distribution of ϕ given X , that is,

$$\hat{\phi}_{Bayes} = E_{\phi|X}[\phi]. \quad (2.3.8)$$

In general cases without conjugacy between prior and posterior distributions, since there is no analytical form of integrals in Eq. 2.3.7, the Monte-Carlo approximation is frequently used to compute $\hat{\phi}_{Bayes}$ as follows

$$\hat{\phi}_{Bayes} \approx \frac{1}{L} \sum_{i=1}^L \hat{\phi}_{(i)} \quad (2.3.9)$$

where $\hat{\phi}_{(i)}$ is a random draw from $\phi|X$ and L is an arbitrary large number.

The result can be easily extended to multivariate parameter cases under the assumption of

conditional independence of parameters, and Eq. 2.3.8 can be rewritten by

$$\hat{\phi}_{i, Bayes} = E_{\phi_i | X, \phi_{-i}}[\phi_i] \quad (2.3.10)$$

where ϕ_i denotes a parameter to estimate and ϕ_{-i} denotes the rest of parameters excluding ϕ_i .

Now suppose that a heterogeneous dataset has K statistically different classes and that each class has its own distinct set of m parameters, that is, $\phi_j = (\phi_{j1}, \dots, \phi_{jm})$ for $j = 1, \dots, K$. Then Eq. 2.3.10 is still valid for a set of parameters, and Bayes estimator for parameter i of set j is

$$\hat{\phi}_{ji, Bayes} = E_{\phi_{ji} | X_j, \phi_{-j}, \phi_{j(-i)}}[\phi_{ji}] \quad (2.3.11)$$

where ϕ_{-j} denotes all sets of parameters except ϕ_j and $\phi_{j(-i)}$ denotes all parameters of ϕ_j except ϕ_{ji} ; X_j is a random variable for class j .

When K is unknown, we can regard K as a latent parameter over ϕ . By making a connection between K and ϕ , K can be also estimated as shown in Eq. 2.3.12. For such a connection, a Dirichlet process [10] is well-formulated and widely used.

$$\hat{K}_{Bayes} = E_{K | X, \phi}[K] \quad (2.3.12)$$

2.4 Adaptive Rejection Sampling

The rejection sampling [11] is a non-Markov simulation sampling technique. In contrast to MCMC methods such as Metropolis-Hastings algorithm and Gibbs sampler, successive

sample points are assumed to be independent, but the way of generating density via rejection inspired many MCMC methods.

Let $f(x)$ be the target density, which has a complex distribution of a non-standard form, and let $h(x)$ be the proposal density, which satisfies $f(x) \leq c \cdot h(x)$ with a known constant c . The proposal density is usually chosen among known standard density functions for easy computing and sampling. The rejection sampling procedure to obtain n points is summarized as follows.

1. Generate a sample observation x_c from $h(x)$
2. Generate a random value $u \sim \mathcal{U}(0, 1)$.
3. If $u \leq \frac{f(x_c)}{c \cdot h(x_c)}$, x_c is accepted
Otherwise, rejected.
4. Repeat 1-3 until n points are accepted.

The choices of $h(x)$ and c are critical in the speed of rejection sampling. The rejection sampling works well when $h(x)$ is a good approximation to $f(x)$. Otherwise, c should be set large and this would result in the large number of rejections. Gilks and Wild [12] pointed out a difficulty of Gibbs sampling in hierarchical modeling. Gibbs sampling requires the full conditional distributions as known standard forms for all parameters, which is not always possible for most of hierarchical models. They proposed the *adaptive rejection sampling* (ARS) as an efficient way to deal with non-conjugacy distributions when the log-density of target distribution function is concave.

Given a log-concave function $g(x)$, let $h(x) = \log g(x)$ for $x \in D$. Define the set of reference points, where derivative is to be computed, T_k by

$$T_k = \{x_i | x_i \leq x_{i+1} \text{ for } i = 1, \dots, k\}, \quad (2.4.1)$$

the point, where two lines tangent to $h(x)$ at x_j and x_{j+1} intersect, z_j by

$$z_j = \frac{h(x_{j+1}) - h(x_j) + x_{j+1}h'(x_{j+1}) + x_j h'(x_j)}{h'(x_j) - h'(x_{j+1})} \quad (2.4.2)$$

for $j = 1, \dots, k - 1$,

the line, tangent to $h(x)$ at $x_j \in T_k$, $u_j(x)$ by

$$u_j(x) = h(x_j) + (x - x_j)h'(x_j) \quad (2.4.3)$$

for $x \in [z_{j-1}, z_j]$ and $j = 1, \dots, k$,

the line, which intersects $h(x)$ at x_j and x_{j+1} , $l_j(x)$ by

$$l_j(x) = \frac{(x_{j+1} - x)h(x_j) + (x - x_j)h(x_{j+1})}{x_{j+1} - x_j} \quad (2.4.4)$$

for $x \in [x_j, x_{j+1}]$ and $j = 1, \dots, k - 1$,

and the area proportion under $u_j(x)$ for $x \in [z_{j-1}, z_j]$, $s_j(x)$ by

$$s_j(x) = \frac{\exp(u_j(x))}{\int_D \exp(u(t)) dt} \cdot \quad (2.4.5)$$

According to the above definitions, the set of function $u_j(x)$ forms an upper-hull, tangent to $h(x)$ at all $x_j \in T_k$, and the set of function $l_j(x)$ forms a lower-hull, tangent to $h(x)$ at all z_j for $j = 1, \dots, k - 1$. Let $U(x)$ and $L(x)$ be those upper- and lower-hulls, respectively. The following condition is always satisfied:

$$L_k(x) \leq h(x) \leq U_k(x). \quad (2.4.6)$$

As k increases, that is, the reference set has more points, upper- and lower-hulls become

more close to $h(x)$:

$$L_k(x) \leq L_{k+1}(x) \leq \dots \leq h(x) \leq \dots \leq U_{k+1}(x) \leq U_k(x). \quad (2.4.7)$$

The algorithm consists of a sampling step and two acceptance tests. At sampling step, a candidate x_c is drawn from $s_j(x)$, and two acceptance tests follow.

1. Generate x_c using $s_j(x)$ for $j = 1, \dots, k$.
(Without loss of generality, assume $x_c \in [x_a, x_{a+1}]$ and $x_c \in [z_{b-1}, z_b]$)
2. Generate $u \sim \mathcal{U}(0, 1)$
3. Squeezing test
If $u \leq \exp(l_a(x_c) - u_b(x_c))$, then accept x_c .
Otherwise, evaluate $h(x_c)$ and $h'(x_c)$.
4. Rejection test
If $u \leq \exp(h(x_c) - u_b(x_c))$, then accept x_c . Otherwise reject x_c .

If the squeezing test fails, this implies that $u_b(x_c)$ may be too large and need to be reduced. In that case, ARS includes x_c to T_k , forms a new set T_{k+1} of reference points, and updates $u_b(x)$, $l_a(x)$, $l_{a+1}(x)$ and $s_b(x)$. By adding more reference points, upper- and low-hulls get close to $h(x)$, and the rate of acceptance will increase in result.

Unlike other sampling techniques including the rejection sampling, ARS does not require a proposal distribution, the choice of which is very sensitive to the rejection rate. Instead of a proposal distribution, upper- and low-hulls of target distribution are adjusted according to the rate of rejection. Since ARS is simple and efficient, it is widely used for sampling non-conjugate distributions [13, 14].

2.5 Dirichlet Process Prior Models

2.5.1 Dirichlet Process

A Dirichlet process (DP) [10] is a stochastic process where the domain of a sample has a random distribution as well as a sample itself.

Definition 2.5.1 *A probability measure $G(\cdot)$ is said to be DP-distributed on the measurable space $(\mathcal{X}, \mathcal{A})$ if the joint probability $(G(A_1), \dots, G(A_K))$ for any finite partition $A_{1:K} \in \mathcal{A}$ of \mathcal{X} has a Dirichlet distribution with the parameter $(\alpha G_0(A_1), \dots, \alpha G_0(A_K))$, and this is denoted by $G(\cdot) \sim DP(G_0(\cdot), \alpha)$.*

Two parameters of a DP, α and $G_0(\cdot)$, are called the concentration parameter and the base distribution, respectively, and α represents how the distribution of a DP concentrates on $G_0(\cdot)$.

Sethuraman [15] showed that a Dirichlet measure is a probability measure on the space of all probability measures and that a DP is discrete with probability one. He also introduced the so-called *stick-breaking* construction, where a random measure $G(\cdot)$ is defined as an infinite sequence

$$G(\cdot) = \sum_{j=1}^{\infty} \pi_j \delta_{\theta_j}(\cdot) \quad (2.5.1)$$

with

$$\begin{aligned} \theta_j &\sim G_0(\cdot) \\ \pi_j &= \beta_j \prod_{i=1}^{j-1} (1 - \beta_i) \\ \beta_j &\sim \mathcal{B}(1, \alpha) \end{aligned} \quad (2.5.2)$$

where $\delta_{\theta_j}(\theta)$ is the Dirac delta function, which is 1 when $\theta = \theta_j$ and 0 otherwise, and \mathcal{B} denotes the beta distribution.

Using the stick-breaking DP prior construction, any distribution can be represented by a mixture of distributions over parameter θ . In his paper [15], it is also shown that the posterior distribution of a DP is a Dirichlet measure as well. This provides the simplicity of posterior parameter sampling. Let $\theta_1, \dots, \theta_n$ be n independent random draws from $G \sim DP(G_0, \alpha)$, and then the posterior distribution of G given $\theta_{1:n}$ is still a Dirichlet process as follows.

$$G|\theta_{1:n} \sim DP\left(\frac{\alpha}{\alpha + K}G_0 + \frac{1}{\alpha + n} \sum_{i=1}^n \delta_{\theta_i}, \alpha + n\right) \quad (2.5.3)$$

As another useful property of a DP, the *Polya urn scheme* [16] can be used to obtain the marginal* distribution of parameter θ_i . Given a statistically exchangeable sequence $\theta_1, \dots, \theta_n$, the marginal distribution of θ_i for $i = 1, \dots, n$ is

$$\theta_i|\theta_{-i} \sim \frac{1}{\alpha + n - 1} \sum_{j:j \neq i} \delta_{\theta_j} + \frac{\alpha}{\alpha + n - 1} G_0 \quad (2.5.4)$$

where θ_{-i} is the set $\{\theta_j\}_{j=1}^n \setminus \theta_i$.

Eq. 2.5.4 clearly shows the clustering effect of a DP. The realization of θ_i is either one of the existing values or a new value generated from the base distribution G_0 .

*Consider the sequence of observations $(\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_n)$ for a random variable θ_i . Then the marginal posterior distribution is $p(\theta_i|\theta_{-i})$.

2.5.2 Dirichlet Process Prior Mixture Models

Let $\mathbf{X} = \{x_1, \dots, x_n\}$ be a sequentially exchangeable sequence from unknown distribution F , that is, x_i , for $i = 1, \dots, n$, is identically and independently drawn from F :

$$x_i \stackrel{\text{iid}}{\sim} F. \quad (2.5.5)$$

For estimating the distribution function $F(x)$, consider the following nonparametric model with parameter $\theta \in \Theta$ which is distributed according to G :

$$F(x) = \int_{\theta \in \Theta} F(x|\theta) dG(\theta) \quad (2.5.6)$$

where $F(\cdot|\theta)$ is the likelihood of distribution F .

In order to obtain a mixture representation of F , assume that G is DP-distributed:

$$G \sim DP(G_0, \alpha), \quad (2.5.7)$$

and substitute Eq. 2.5.4 for G in Eq. 2.5.6. The resulting distribution function F is an infinite countable mixture of likelihood functions

$$F(x) = \frac{1}{\alpha + K - 1} \sum_{j=1}^K F(x|\phi_j) + \frac{\alpha}{\alpha + K - 1} \int F(x|\theta) dG_0(\theta) \quad (2.5.8)$$

where ϕ_j is a distinct parameter such that $\phi_j \in \Theta$ and $\phi_j \neq \phi_i$ for $j \neq i$, and K is the number of mixture components.

In summary, a nonparametric Dirichlet process prior mixture model can be described as

follows.

$$\begin{aligned}
x_i|\theta_i &\sim F(x_i|\theta_i) \\
\theta_i|G &\sim G \\
G &\sim DP(G_0, \alpha)
\end{aligned}
\tag{2.5.9}$$

where θ_i is the parameter for observation x_i for $i = 1, \dots, n$.

By introducing the set of class-indicator variables $c = \{c_1, \dots, c_n\}$, where c_i denotes the class identity of x_i , and explicating the number of mixture components, denoted by K , a finite equivalent form of Eq. 2.5.9 can be written by

$$\begin{aligned}
x_i|c_i, \phi &\sim F(\phi_{c_i}) \\
c_i|\pi &\sim Discrete(\pi_1, \dots, \pi_K) \\
\phi_c &\sim G_0 \\
\pi &\sim \mathcal{D}(\alpha/K, \dots, \alpha/K)
\end{aligned}
\tag{2.5.10}$$

where ϕ_i , for $i = 1, \dots, K$, is the parameter of mixture component i , $\pi = \{\pi_1, \dots, \pi_K\}$ is a positive vector of mixing proportions, and \mathcal{D} denotes the Dirichlet distribution. Here $F(\cdot|\theta_i)$ is a distribution function notation and $F(\phi_{c_i})$ is a distribution notation, and they are equivalent. (for instance, $x \sim \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}) \equiv x \sim \mathcal{N}(0, 1)$)

The prior distribution of c_i can be easily obtained by integrating out π in the joint probability distribution $p(c_i, \pi)$ and then taking the limit of K at infinity (see [17] for detailed

derivation), and the result is

$$P(c_i = c^* | c_{-i}, \alpha) = \begin{cases} \frac{n_{c^*}}{\alpha + n - 1} & \text{if } c^* \in c_{-i} \quad \text{for } i = 1, \dots, K \\ \frac{\alpha}{\alpha + n - 1} & \text{if } c^* \notin c_{-i} \quad \text{for any } i \in \{1, \dots, K\} \end{cases} \quad (2.5.11)$$

where n_{c^*} is the number of observations excluding x_i in the mixture component c^* , and c_{-i} is the set of indicators excluding c_i . Therefore c_i has the same form of distribution as θ_i in Eq. 2.5.4 as follows.

$$c_i | c_{-i}, \alpha \sim \frac{1}{\alpha + n - 1} \sum_{j: j \neq i} \delta_{c_j} + \frac{\alpha}{\alpha + n - 1} G_0 \quad (2.5.12)$$

Let ϕ_{all} be the set of parameters of all mixture components. For $c^* \in c_{-i}$ and $c^{**} \notin c_{-i}$, the posterior probabilities of c_i are respectively defined as follows.

$$\begin{aligned} P(c_i = c^* | c_{-i}, x_i, \phi_{all}, \alpha) &= P(c_i = c^* | c_{-i}, x_i, \phi_{c^*}, \alpha) \\ &\propto P(c_i = c^* | c_{-i}, \alpha) F(x_i | \phi_{c^*}) \\ &= \frac{n_{c^*}}{\alpha + n - 1} F(x_i | \phi_{c^*}) \end{aligned} \quad (2.5.13)$$

$$\begin{aligned} P(c_i = c^{**} | c_{-i}, x_i, \phi_{all}, \alpha) &= P(c_i = c^{**} | c_{-i}, x_i, \alpha) \\ &\propto P(c_i = c^{**} | c_{-i}, \alpha) F(x_i) \\ &= P(c_i = c^{**} | c_{-i}, \alpha) \int F(x_i | \phi) dG_0(\phi) \\ &= \frac{\alpha}{\alpha + n - 1} \int F(x_i | \phi) dG_0(\phi) \end{aligned} \quad (2.5.14)$$

The above two equations shows how mixture components are formed and created. The

first terms of two equations follow the Polya's urn scheme, that is, as more populated, a component is more likely to be large. The second term of Eq. 2.5.13 is how fit x_i is to component c^* , which is characterized by parameter ϕ_{c^*} . Therefore, the component membership of x_i is determined by the product of the size of mixture and the likelihood F . In case that a new component is created by Eq. 2.5.14, all other possibilities of ϕ is considered and this is easily represented by an integral form. However, the computation of the integral may not be analytically feasible when G_0 is not conjugate prior for F .

Escobar [18], followed by MacEachern and Müller [19], proposed an approximation using Gibbs sampling to compute the non-conjugate integral:

$$\int F(x_i|\phi)dG(\phi) \approx \frac{1}{M} \sum_{j=1}^M F(x_i|\phi_{(j)}) \quad (2.5.15)$$

where $\phi_{(j)}$, for $j = 1, \dots, M$, is a realization of parameter ϕ , distributed from G , and M is an arbitrary large number.

Neal [17] provided another method, based on Metropolis-Hastings algorithm, to avoid integrals for non-conjugate priors. He pointed out a potential problem of approximations, based on Gibbs sampling, in a multidimensional space, where the probability that a new component will be chosen is very low when the size of samples, M , is not large enough. He proposed that, rather than using Eq. 2.5.13 and Eq. 2.5.14 directly, c_i is sampled according to Eq. 2.5.12, for $i = 1, \dots, n$, with ϕ_i sampled from G_0 for newly created components, and a Metropolis-Hastings ratio test is performed in terms of $F(\cdot|\phi_{c_i}^{\text{new}})$ against $F(\cdot|\phi_{c_i}^{\text{prev}})$ where $\phi_{c_i}^{\text{new}}$ and $\phi_{c_i}^{\text{prev}}$ are the new and previous parameters for component c_i , respectively. The approach only performs pairwise-comparisons between previous and new parameters once a new component index is sampled. Thus, newly-created parameters, which might be removed by a Gibbs sampling approximation because of under-estimation for an integral, would

survive more likely. He also provided its modification by adding a multiplicative term to the ratio test in order to guarantee ergodic Markov chain movement.

From Eq. 2.5.12 and Eq. 2.5.14, it is easily noticed that the parameter α has significant effects on the creation of components. When α is much less than the cardinality n_j of mixture components (that is, $\alpha \ll n_j, \forall j$), it is less likely to create a new component. On the other hand, when $\alpha \gg n_j, \forall j$, the number of components would increase fast. Antoniak [20] showed $E(K) \approx \alpha \log((n + \alpha)/\alpha)$, and Escobar [18] and Richardson *at al.* [21] verified the sensitivity of K in terms of α and n . Therefore, it is necessary to update the concentration parameter α properly in terms of the cardinalities of components for insensitive Dirichlet process prior models to the number of observations. With inverse- χ^2 as a vague (or flat) prior for α , the likelihood and the posterior distribution for α can be derived as follows.

$$\begin{aligned} P(n_{1:K}|\alpha) &= \frac{\alpha^K \Gamma(\alpha)}{\Gamma(n + \alpha)} \\ P(\alpha|n_{1:K}) &\propto \frac{\alpha^{K-\frac{3}{2}} e^{-\frac{1}{2\alpha}} \Gamma(\alpha)}{\Gamma(n + \alpha)} \end{aligned} \quad (2.5.16)$$

Example 2.5.2 *Infinite Gaussian mixture models*

Suppose that x is a d -dimensional random vector generated from a mixture of K Gaussian distributions with parameters (μ_j, s_j^{-1}) where K is a parameter denoting the number of mixture components, μ_j is a d -dimensional mean vector and s_j is a d -by- d precision matrix (inverse of covariance matrix), for $j = 1, \dots, K$. This can be written as

$$x \sim \sum_{j=1}^K \pi_j \mathcal{N}(\mu_j, s_j^{-1}) \quad (2.5.17)$$

where μ_j, s_j^{-1} and π_j are the mean, the precision and the mixing proportion for component j , respectively for $j = 1, \dots, K$, and the number of component, K , is unknown.

Suppose that the random vector X_j for component j is drawn from a Gaussian distribution with parameter (μ_j, s_j^{-1}) , that is,

$$X_j | \mu_j, s_j \sim \mathcal{N}(\mu_j, s_j^{-1}). \quad (2.5.18)$$

Introducing hyperparameters λ and r , let a component mean vector μ_j has a Gaussian distribution for conjugacy as

$$\mu_j | \lambda, r \sim \mathcal{N}(\lambda, r^{-1}) \quad (2.5.19)$$

where hyperparameters have conformed dimensionality.

A precision parameter s_j can be also represented with its conjugate prior (Wishart distribution, denoted by \mathcal{W}) by introducing hyperparameters, ρ and w , as follows.

$$s_j | \rho, w \sim \mathcal{W}(\rho, \frac{1}{\rho} w^{-1}). \quad (2.5.20)$$

Hierarchical dependency structure of a Gaussian mixture model between parameters is shown in Figure 2.5.1.

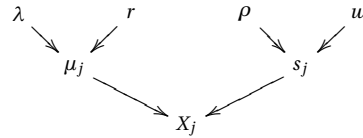


Figure 2.5.1: The conditional dependency of parameters in Gaussian mixture models

For a statistically exchangeable observation sequence $\mathbf{X} = \{x_1, \dots, x_n\}$ with μ_x and σ_x^2 as a

mean vector and a covariance matrix, respectively, prior distributions for hyperparameters have the following uninformative forms to be flat enough.

$$\begin{aligned}
\lambda &\sim \mathcal{N}(\mu_x, \sigma_x^2), \\
r &\sim \mathcal{W}(d, \sigma_x^{-2}), \\
\rho &\sim \chi^{-2}(1), \\
w &\sim \mathcal{W}(1, \sigma_x^2)
\end{aligned} \tag{2.5.21}$$

where $\chi^{-2}(c)$ denotes the inverse of χ^2 with c degree of freedom.

Due to the conjugacy applied for parameter priors, the posterior distributions of parameters can be easily obtained. Let c_i be an index variable denoting the mixture membership of x_i , that is, $c_i = j$ means that x_i is drawn from component j which has a Gaussian distribution $\mathcal{N}(\mu_j, s_j^{-1})$.

The posterior distribution of μ_j is

$$\mu_j | c_{1:n}, \mathbf{X}, s_j, \lambda, r \sim \mathcal{N} \left\{ (r + n_j s_j)^{-1} (r \lambda + n_j s_j \bar{X}_j), (r + n_j s_j)^{-1} \right\} \tag{2.5.22}$$

where n_j and \bar{X}_j denote the number of observations and the arithmetic sample mean vector of observations in component j .

For s_j , its posterior distribution is also a Wishart distribution like its prior due to conjugacy, and given by

$$s_j | c_{1:n}, \mathbf{X}, \mu_j, \rho, w \sim \mathcal{W} \left\{ \rho + n_j, \left[w \rho + \sum_{i:c_i=j} (x_i - \mu_j)(x_i - \mu_j)' \right]^{-1} \right\}. \tag{2.5.23}$$

Similarly as μ_j ,

$$\lambda | \mu_{1:K}, r \sim \mathcal{N} \left\{ (\sigma_x^{-2} + Kr)^{-1} \left(\sigma_x^{-2} \mu_x + r \sum_{j=1}^K \mu_j \right), (\sigma_x^{-2} + Kr)^{-1} \right\}. \quad (2.5.24)$$

The posterior distributions for hyperparameter r and w are as follows.

$$r | \mu_{1:K}, \lambda \sim \mathcal{W} \left\{ d + K, \left[d \sigma_x^2 + \sum_{j=1}^K (\mu_j - \lambda)(\mu_j - \lambda)' \right]^{-1} \right\} \quad (2.5.25)$$

$$w | s_{1:K}, \rho \sim \mathcal{W} \left\{ K\rho + d, \left[d \sigma_x^{-2} + \rho \sum_{j=1}^K s_j \right]^{-1} \right\} \quad (2.5.26)$$

Since the posterior distributions derived above are of standard form, each parameter can be easily sampled using Gibbs sampler. However, the hyperparameter ρ is not the case as follows.

$$\begin{aligned} p(\rho | s_{1:K}, w) &\propto p(\rho) p(s_{1:K} | \rho, w) \\ &= p(\rho) \prod_{j=1}^K p(s_j | \rho, w) \\ &= \rho^{-\frac{3}{2}} \exp\left(-\frac{1}{2\rho}\right) \prod_{j=1}^K \frac{|s_j|^{\frac{\rho-d-1}{2}}}{2^{\frac{\rho d}{2}} \left|\frac{w^{-1}}{\rho}\right|^{\frac{\rho}{2}} \Gamma_d\left(\frac{\rho}{2}\right)} \exp\left\{-\frac{1}{2} \text{Tr}(\rho w s_j)\right\} \end{aligned} \quad (2.5.27)$$

where $\Gamma_p(\cdot)$ is the multivariate gamma function and $\text{Tr}(\cdot)$ is the trace of a square matrix.

Using Eq. 2.5.16, the posterior distribution of concentration parameter α of Dirichlet process

has the following form.

$$\begin{aligned}
p(\alpha | K, n_{1:K}) &\propto p(n_{1:K} | \alpha) p(\alpha) \\
&= \alpha^{K-2} \exp\left(-\frac{1}{\alpha}\right) \frac{\Gamma(\alpha)}{\Gamma(n + \alpha)}
\end{aligned} \tag{2.5.28}$$

As shown above, hyperparameters ρ and α have posterior distributions of non-standard form, and hence Gibbs sampling is not applicable. Rasmussen [22] mentioned that those distributions have log-concavity, and suggested to use the adaptive rejection sampling (ARS) [12]. He described formulas in detail for univariate parameters and explained possible extension to multivariate cases.

By Eq. 2.5.13, the posterior probability that c_i changes to one of existing components is given by, for $j = 1, \dots, K$,

$$\begin{aligned}
p(c_i = j | c_{-i}, \mu_j, s_j, \alpha) &\propto p(c_i = j | c_{-i}, \alpha) p(x_i | \mu_j, s_j) \\
&\propto \frac{n_{j,-i}}{n - 1 + \alpha} \phi(x_i | \mu_j, s_j).
\end{aligned} \tag{2.5.29}$$

The other case, when a new component is created, is given by Eq. 2.5.30.

$$\begin{aligned}
&p(c_i \neq j, \forall j \in c_{-i} | c_{-i}, \lambda, r, \rho, w, \alpha) \\
&\propto p(c_i \neq j, \forall j \in c_{-i} | c_{-i}, \alpha) \int p(x_i | \mu, s) p(\mu | \lambda, r) p(s | \rho, w) d\mu ds \\
&\propto \frac{\alpha}{n - 1 + \alpha} \int \phi(x_i | \mu, s) \phi(\mu | \lambda, r) p(s | \rho, w) d\mu ds
\end{aligned} \tag{2.5.30}$$

2.5.3 Dirichlet Process Prior Regression Models

While DP mixture models have density estimation (parameter estimation) [13, 14, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27] or clustering [23, 25, 28] as the main purpose, the response estimation (prediction) [7, 8, 9, 23, 25, 29] is the main aim of DP prior regression models. Since clustering is a *by-product* of models with DP prior, the inference of clustering (or classification) is a common characteristic of both DP mixture models and DP prior regression models.

A DP prior regression model is a nonparametric regression technique, which can handle heterogeneous population data in the sense that the number of classes is drawn from data. Its prediction is robust because a Bayesian model-averaging estimate over completing models accommodates the variation of predictions and avoids overfitting. A DP prior regression model is also flexible in that the degree of heterogeneity does not need to be decided before modeling.

In contrast to DP mixture models, a DP prior regression model has a response y , expressed in terms of covariate x and parameter θ . The parameter θ is decomposed to two independent parameters, θ_x and θ_y , related with x and y , respectively. Thus, the distribution of $y|x$ can be represented in a nonparametric way with the decomposed parameter $\theta = (\theta_x, \theta_y)$ as

$$\begin{aligned} F(y|x) &= \int F(y|x, \theta) dG(\theta) \\ &= \int F_y(y|x, \theta_y) F_x(x|\theta_x) dG(\theta) \end{aligned} \tag{2.5.31}$$

where F, G, F_x and F_y denote the distribution functions or likelihood functions of their own arguments.

Applying Dirichlet process (DP) prior to θ as the following

$$\begin{aligned}\theta &\sim G \\ G &\sim DP(G_0, \alpha),\end{aligned}\tag{2.5.32}$$

the *stick-breaking* construction makes θ have a discrete distribution with probability 1 as follows

$$G(\theta) \sim \sum_{j=1}^{\infty} \pi_j \delta_{\pi_j}(\theta).\tag{2.5.33}$$

Due to the DP prior, the posterior distribution of θ also follows DP distribution as

$$G|\theta_{1:K} \sim DP\left(\frac{1}{\alpha + K} \sum_{j=1}^K \delta_{\theta_j} + \frac{\alpha}{\alpha + K} G_0, \alpha + n\right).\tag{2.5.34}$$

By the *Polya urn scheme*, the marginal distribution of θ is

$$G(\theta_i|\theta_{-i}) \sim \frac{1}{\alpha + K - 1} \sum_{j=1}^K \delta_{\theta_j} + \frac{\alpha}{\alpha + K - 1} G_0,\tag{2.5.35}$$

and Eq. 2.5.31 has a mixture representation

$$F(y|x) = \frac{1}{\alpha + K - 1} \sum_{j=1}^K F_y(y|x, \theta_{y,j}) F_x(x|\theta_{x,j}) + \frac{\alpha}{\alpha + K - 1} \int F_y(y|x, \theta_y) F_x(x|\theta_x) dG_0(\theta).\tag{2.5.36}$$

where $\theta_{x,j}$ and $\theta_{y,j}$ are parameters related to x and y of component j , respectively, and K denotes the number of mixture components.

In summary, a DP prior regression model can be written as follows.

$$\begin{aligned}
y_i|x_i, \theta_{y_i} &\sim F_y(y_i|x_i, \theta_{y_i}) \\
x_i|\theta_{x_i} &\sim F_x(x_i|\theta_{x_i}) \\
\theta_i = (\theta_{x_i}, \theta_{y_i})|G &\sim G \\
G &\sim DP(G_0, \alpha)
\end{aligned} \tag{2.5.37}$$

Similar to Eq. 2.5.10 of Dirichlet process mixture model, by introducing indicator variables, the finite equivalent form can be derived as

$$\begin{aligned}
y_i|x_i, c_i, \phi &\sim F_y(x_i, \phi_{y,c_i}) \\
x_i|\phi &\sim F_x(\phi_{x,c_i}) \\
c_i|\pi &\sim Discrete(\pi_1, \dots, \pi_K) \\
\phi_c &\sim G_0 \\
\pi &\sim \mathcal{D}(\alpha/K, \dots, \alpha/K)
\end{aligned} \tag{2.5.38}$$

where ϕ_{y,c_i} and ϕ_{x,c_i} denote the parameters of component c_i associated with x and y , respectively.

Let ϕ_{all} be the set parameters of all current mixture components as in Eq. 2.5.13. For $c^* \in c_i$ and $c^{**} \notin c_i$, their posterior distributions can be drawn similarly as follows.

$$\begin{aligned}
P(c_j = c^*|c_{-i}, y_i, x_i, \phi_{all}, \alpha) &= P(c_i = c^*|c_{-i}, y_i, x_i, \phi_{c^*}) \\
&\propto P(c_i = c^*|c_{-i}, \alpha) F_x(x_i|\phi_{x,c^*}) F_y(y_i|\phi_{y,c^*}) \\
&= \frac{n_{c^*}}{\alpha + n - 1} F_x(x_i|\phi_{x,c^*}) F_y(y_i|\phi_{y,c^*})
\end{aligned} \tag{2.5.39}$$

$$\begin{aligned}
P(c_i = c^{**} | c_{-i}, y_i, x_i, \phi_{all}, \alpha) &= P(c_i = c^{**} | c_i, y_i, x_i, \alpha) \\
&\propto P(c_i = c^{**} | c_i, \alpha) P(y_i | x_i) P(x_i) \\
&= P(c_i = c^{**} | c_i, \alpha) P(y_i | x_i) P(x_i) \\
&= \frac{\alpha}{\alpha + n - 1} \int F_y(y_i | x_i, \phi_y) F_x(x_i | \phi_x) dG_0(\phi) \quad (2.5.40)
\end{aligned}$$

The rest of formulas and descriptions are exactly the same as DP mixture models in Section 2.5.2.

Given a value of covariate x , the response estimate is computed by

$$\begin{aligned}
E(y|x) &= \int f_y(x, \theta) F(y|x, \theta) dG(\theta) \\
&= \frac{1}{b} \sum_{j=1}^K y_j F_y(y_j | x, \theta_{y,j}) F_x(x | \theta_{x,j}) + \frac{\alpha}{b} \int f_y(x, \theta) F_y(y|x, \theta_y) F_x(x | \theta_x) dG_0(\theta) \quad (2.5.41)
\end{aligned}$$

where f_y is the relationship function of response, y_j is the response estimate of mixture component j (component-wise estimate, that is, $y_j = f_y(x, \theta_j)$), and b is the probability normalizing constant as follows

$$b = \sum_{j=1}^K F_y(y_j | x, \theta_{y,j}) F_x(x | \theta_{x,j}) + \alpha \int F_y(y|x, \theta_y) F_x(x | \theta_x) dG_0(\theta). \quad (2.5.42)$$

Example 2.5.3 *Dirichlet Process Linear Models (DPLMs)*

Suppose a covariate-response random variable pair (x, y) has a mixture of linear relationships with proportion parameters π_j and location-scale parameters $\mathcal{N}([1, x']\beta_j, s_{y,j}^{-1})$ for $j = 1, \dots, K$, where β_j is a regression coefficient vector for component j and K is an unknown parameter

denoting the number of components, that is,

$$y|x \sim \sum_{j=1}^K \pi_j \mathcal{N} \left([1, x'] \beta_j, s_{yj}^{-1} \right). \quad (2.5.43)$$

Also, suppose that covariate X_j for mixture component j is drawn from a Gaussian distribution with parameter (μ_j, s_{xj}^{-1}) as:

$$X_j | \mu_j, s_{xj} \sim \mathcal{N}(\mu_j, s_{xj}^{-1}). \quad (2.5.44)$$

As done in Example 2.5.2, hyperparameters, λ and r , are introduced and formulated to utilize conjugacy.

$$\begin{aligned} \mu_j | \lambda, r &\sim \mathcal{N}(\lambda, r^{-1}) \\ s_{xj} | \rho, w &\sim \mathcal{W}(\rho, \frac{1}{\rho} w^{-1}) \end{aligned} \quad (2.5.45)$$

Similarly, as for Y_j , the response for mixture component j , its likelihood and the prior hyperparameter distribution are set up as follows.

$$\begin{aligned} Y_j | x_j, \beta_j, s_{yj} &\sim \mathcal{N} \left([1, x_j'] \beta_j, s_{yj}^{-1} \right) \\ \beta_j | m, v &\sim \mathcal{N}(m, v^{-1}) \\ s_{yj} | a, \theta &\sim \mathcal{G}(a, \frac{1}{a\theta}) \end{aligned} \quad (2.5.46)$$

Therefore, the conditional independence of parameters in a Dirichlet process linear model is represented by Figure 2.5.2. Given covariate-response pairs $(\mathbf{X}, \mathbf{Y}) = \{(x_i, y_i)\}_{i=1}^n$, uninformative priors for hyperparameters are set such that they are flat enough, and a candidate set of

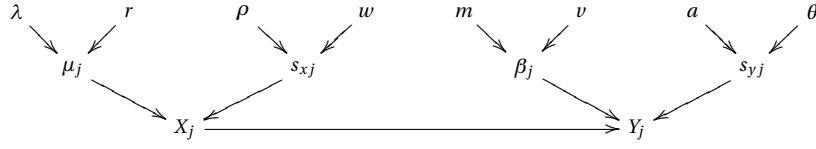


Figure 2.5.2: The conditional dependency of parameters in DPLMs

priors would be the followings.

$$\begin{aligned}
 \lambda &\sim \mathcal{N}(\mu_x, \sigma_x^2), \\
 r &\sim \mathcal{W}(d, \sigma_x^{-2}), \\
 \rho &\sim \chi^{-2}(1), \\
 w &\sim \mathcal{W}(1, \sigma_x^2) \\
 m &\sim \mathcal{N}(B_{init}, \sigma_{init}^2) \\
 v &\sim \mathcal{W}(d+1, \sigma_{init}^{-2}) \\
 a &\sim \chi^{-2}(1) \\
 \theta &\sim \mathcal{G}(1, \sigma_e^2)
 \end{aligned} \tag{2.5.47}$$

where σ_x^2 is the covariance matrix of \mathbf{X} , d is the dimension of covariate, B_{init} is the zero vector of $(d+1)$ dimensionality, and σ_e^2 is the variance of residuals from the least squared solution of (\mathbf{X}, \mathbf{Y}) . For computing σ_{init}^2 , assume that there is a component, which contains all observations, and z is the n -by- $(d+1)$ design matrix of the component. Thus, the response vector y can be expressed as $y = z\beta + \epsilon$ with a disturbance term ϵ . Then $var(\beta)$ can be

computed in the following way.

$$\begin{aligned}
z'y &= z'z\beta + z'\epsilon \\
\Rightarrow z'z\beta &= z'y - z'\epsilon \\
\Rightarrow \beta &= (z'z)^{-1}z'y - (z'z)^{-1}z'\epsilon \\
\Rightarrow \beta &= b - (z'z)^{-1}z'\epsilon \\
&\text{(} b \text{ is the least square solution of linear regression } y = z\beta \text{)} \\
\Rightarrow \frac{1}{n}(\beta - b)(\beta - b)' &= \frac{1}{n}\{(z'z)^{-1}z'\epsilon\}\{(z'z)^{-1}z'\epsilon\}' \\
\Rightarrow \text{var}(\beta) &= \frac{1}{n}\{(z'z)^{-1}z'\epsilon\}\{(z'z)^{-1}z'\epsilon\}' \tag{2.5.48}
\end{aligned}$$

Therefore, σ_{init}^2 can be safely set to $\text{var}(\beta)$ in Eq. 2.5.48, and the remaining σ_e^2 is simply $\text{var}(y - zb)$. The posterior distribution can be derived easily because of conjugacy applied to the selection of priors. The resulting posterior distributions for parameters are listed in Table 2.5.3.

Table 2.5.1: The prior distributions and corresponding posteriors for parameters in Dirichlet process linear models where B_j is the least square solution for β_j , i.e., $Y_j = X_j' B_j$.

Prior distribution	Posterior distribution
$Y_j x_j, s_{yj}, \beta_j \sim \mathcal{N}([1 : x_j'] \beta_j, s_{yj}^{-1})$	\cdot
$X_j \mu_j, s_{xj} \sim \mathcal{N}(\mu_j, s_{xj}^{-1})$	\cdot
$\mu_j \sim \mathcal{N}(\lambda, r^{-1})$	$\mu_j c, x, s_{xj}, \lambda, r \sim \mathcal{N}((r + n_j s_{xj})^{-1}(r\lambda + n_j s_{xj} \bar{y}_j), (r + n_j s_{xj})^{-1})$
$s_{xj} \sim \mathcal{W}(\rho, \frac{1}{\rho} w^{-1})$	$s_{xj} c, x, \mu_j, \rho, w \sim \mathcal{W}(\rho + n_j, [\rho \beta_j + \sum_{\{i: c_i=j\}} (x_i - \mu_j)(x_i - \mu_j)']^{-1})$
$\beta_j \sim \mathcal{N}(m, v^{-1})$	$\beta_j m, v, y_{i \in c_j} \sim \mathcal{N}((v + s_{yj} W_j)^{-1}(v m + s_{yj} w_j B_j), (v + s_{yj} w_j)^{-1})$
$S_{Y_j} \sim \mathcal{G}(a, \frac{1}{a\theta})$	$s_{yj} c, y_{i \in c_j}, \beta_j, a, \theta \sim \mathcal{G}(a + n_j, [a\theta + (B_j - \beta_j)' w (B_j - \beta_j)]^{-1})$
$\lambda \sim \mathcal{N}(\mu_x, \sigma_x^2)$	$\lambda \mu_{1:k}, r \sim \mathcal{N}((\sigma_x^{-2} + kr)^{-1}(\sigma_x^{-2} \mu_x + r \sum_{j=1}^k \mu_j), (\sigma_x^{-2} + kr)^{-1})$
$r \sim \mathcal{W}(d, \sigma_x^{-2})$	$r \mu_{1:k}, \lambda \sim \mathcal{W}(d + k, [d\sigma_x^2 + \sum_{j=1}^k (\mu_j - \lambda)(\mu_j - \lambda)']^{-1})$
$\rho \sim \chi^{-2}(1)$	$p(\rho s_{x1:k}, w) \propto \rho^{-\frac{3}{2}} \exp(-\frac{1}{2\rho}) \prod_{j=1}^K \frac{ s_{xj} ^{\frac{\rho-d-1}{2}}}{2^{\frac{\rho d}{2}} \frac{w^{-1}}{\rho} ^{\frac{\rho}{2}} \Gamma_d(\frac{\rho}{2})} \exp\{-\frac{1}{2} Tr(\rho w s_{xj})\}$ (adaptive rejection sampling)
$w \sim \mathcal{W}(1, \sigma_x^2)$	$w s_{x1:k}, \rho \sim \mathcal{W}(k\rho + 1, [\sigma_x^{-2} + \beta \sum_{j=1}^k s_{xj}]^{-1})$
$m \sim \mathcal{N}(B_{init}, \sigma_{init}^2)$	$m \sim \mathcal{N}((\sigma_{init}^{-2} + kv)^{-1}(\sigma_{init}^{-2} B_{init} + v \sum_{j=1}^k \beta_j), (\sigma_{init}^{-2} + kv)^{-1})$
$v \sim \mathcal{W}(d + 1, \sigma_{init}^{-2})$	$v \sim \mathcal{W}(d + k + 1, [d\sigma_{init}^2 + \sum_{j=1}^k (\beta_j - m)(\beta_j - m)']^{-1})$
$a \sim \chi^{-2}(1)$	$p(a s_{y1:k}, \theta) \propto \rho^{-\frac{3}{2}} \exp(-\frac{1}{2a}) \prod_{j=1}^K \frac{(s_{yj})^{\frac{a}{2}-1}}{2^{\frac{a}{2}} (a\theta)^{-\frac{a}{2}} \Gamma(\frac{a}{2})} \exp(-\frac{1}{2} a\theta s_{yj})$ (adaptive rejection sampling)
$\theta \sim \mathcal{G}(1, \sigma_e^2)$	$\theta s_{y1:k} \sim \mathcal{G}(1 + ka, [\sigma_e^{-2} + a \sum_{j=1}^k s_{yj}]^{-1})$
$\alpha \sim \chi^{-2}(1)$	$p(\alpha K, n_{1:K}) \propto \alpha^{K-2} \exp(-\frac{1}{\alpha}) \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)}$ (adaptive rejection sampling)

Multilevel Dirichlet Process Linear Models

Since a *Dirichlet process linear model* (DPLM) assumes that the relationship between covariate and response is linear and the random error of each mixture component is independent and identically distributed (*i.i.d.*), it has difficulty in modeling data where the response is not a linear function of covariate or the variance components are of either heteroscedasticity or interaction. As a nonlinear extension of DPLM, a *Dirichlet process generalized linear model* (DPGLM) [7, 8] allows nonlinear relationship functions between the linear predictor and the mean of response through the inverse of link function as a *generalized linear model* (GLM) [2, 30]. In addition, by incorporating variance components into the linear predictor, DPGLM can handle heteroscedasticity. However, the choice of forms of covariate-response function (or distribution) and variance component requires human efforts, such as data analysis (or mining) and experts' knowledge, throughout possible candidates, and random errors of mixture components are still assumed to independent. As another extension of DPLM, we

propose the multilevel Dirichlet process linear model (MDPLM) to overcome the deficiencies of DPLM and DPGLM while keeping the simple assumptions of DPLM.

3.1 Description

Our model MDPLM is based on finding hidden random effects which result from discrepancy between the data assumption of a model and the distribution of data. In this section, we will briefly introduce the basic idea of MDPLM.

Suppose that we want to make a model to represent the distribution of a population of covariate-response (x, y) . With a selection of model M , the population distribution of (x, y) can be represented by

$$p(y|x) = p(y|x, M) + \delta_M(x) \quad (3.1.1)$$

where $p(y|x)$ is the population distribution; $p(y|x, M)$ is the population distribution given a model M ; $\delta_M(x)$ is the error distribution due to the difference of data assumptions of M from the population distribution.

Suppose that M_o is another model which overfits a sampled dataset D from the population and has the same complexity as M . Then the distribution can be also represented in terms of model M_o and its error distribution $\delta_{M_o}(x)$ as follows.

$$p(y|x) = p(y|x, M_o) + \delta_{M_o}(x) \quad (3.1.2)$$

Due to the overfitting, the prediction error, decomposed to bias and variance terms, is larger

in M_o than M and the relationship of errors between two models can be written by

$$\text{var}[\delta_M(x)] \leq \text{var}[\delta_{M_o}(x)]. \quad (3.1.3)$$

However, since the bias is less in M_o , the variance for the observable dataset D is also less in M_o as follows.

$$\text{var}[\delta_{M_o}(x)] \leq \text{var}[\delta_M(x)] \quad \text{if } (x, y) \in D \quad (3.1.4)$$

Therefore, given M , M_o and D , we draw the direction of model improvement from the above equation and generate a hidden variable x_e associated with the direction. Suppose that x_e has the distribution of $F(x|M, M_o)$. Then the distribution of (x, y) can be written by

$$p(y|x) = p(y|x, M) + p(\delta_M|x, x_e) + \delta_{M, M_o}(x, x_e) \quad (3.1.5)$$

$$x_e \sim F(x|M, M_o)$$

where $\delta_{M, M_o}(x, x_e)$ is the error distribution after modeling δ_M with x and x_e .

If the variance terms of Eq. 3.1.1 and Eq. 3.1.5 have the relationship

$$\text{var}[\delta_{M, M_o}(y|x)] \leq \text{var}[\delta_M(y|x)] \quad (3.1.6)$$

for the population (x, y) as well as D , then the model described by Eq. 3.1.5 will be better than one by Eq. 3.1.1 because of the lower prediction error.

For the sake of simplicity, we rewrite Eq. 3.1.5 by combining the first two terms of r.h.s. as follows.

$$p(y|x) = p(y|x, x_e, M, M_o) + \delta_{M, M_o}(x, x_e) \quad (3.1.7)$$

Now we serialize the process by letting the superscript (ℓ) denote the association with the ℓ th modeling as follows.

$$p(y|x) = p\left(y|x, x_e^{(0:\ell)}, M^{(0:\ell)}, M_o^{(0:\ell)}\right) + \delta_{M^{(0:\ell)}, M_o^{(0:\ell)}}\left(x, x_e^{(\ell)}\right) \quad (3.1.8)$$

$$x_e^{(i)} \sim F(x|M^{(i)}, M_o^{(i)}) \quad \text{for } i = 0, \dots, \ell.$$

Therefore, the optimal model in the framework is obtained by solving the following problem.

$$\min_{\ell, M^{(0:\ell)}, M_o^{(0:\ell)}} \text{var} \left[\delta_{M^{(0:\ell)}, M_o^{(0:\ell)}}\left(x, x_e^{(\ell)}\right) \right] \quad (3.1.9)$$

Within the framework described above, a complex distribution can be modeled with a series of simple modeling. The proposed MDPLM has layers each of which corresponds to a modeling process of the series in the above framework, and each layer has a mixture of linear models as data assumption. Therefore, MDPLM starts from the simplest form and adds complexity by increasing layers.

3.2 Structure of Multilevel Dirichlet Process Linear Model

MDPLM consists of DPLM layers, each of which has two distinct DPLMs (a generic DPLM and an overfit DPLM) except the last layer as shown in Figure 3.2.1. While a generic DPLM, simply denoted by DPLM, of a layer is equivalent to DPLM, an overfit DPLM, denoted by oDPLM, provides overfit estimates which will be used to generate a hidden random effect. Assuming that every layer is conditionally independent and random errors are also *i.i.d.* in class given covariate, the cumulative estimate of response y at the layer L given a sequence

of covariates is simply the sum of estimates of layers as follows.

$$E(y|\mathbf{X}^{(0)}, \dots, \mathbf{X}^{(L)}) = \hat{y} + \sum_{\ell=0}^{L-1} \hat{e}_\ell \quad (3.2.1)$$

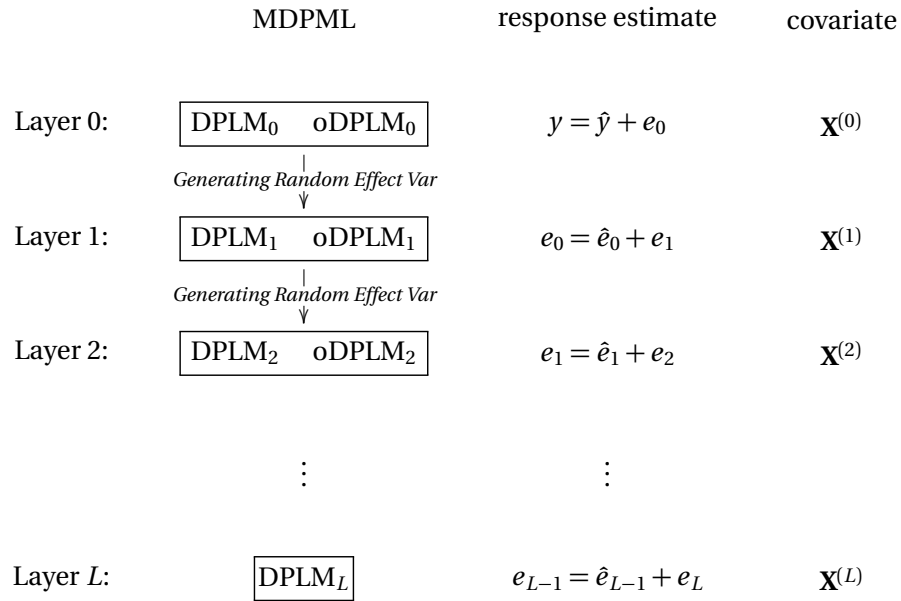


Figure 3.2.1: The schematic of MDPLM: \hat{y} and \hat{e}_ℓ are layer-wise estimates for y and e_ℓ , respectively; e_ℓ and $\mathbf{X}^{(\ell)}$ are the error component and the extended covariate of layer ℓ for $\ell = 0, \dots, L$, respectively.

3.3 Covariate Extension of Layers

A simple description of MDPLM is a sequential error-estimate model where the error component of a layer is estimated in the next layer. If every layer used the same covariate, the

variance of error would not change and this can be easily shown because DPLM assumes that the error component of a class is *i.i.d.*

$$var(e_\ell) = var(e_{\ell-1} | \mathbf{X}^{(\ell-1)}) = var(e_{\ell-1}) \quad \text{for } \ell = 1, \dots, L \quad (3.3.1)$$

Hence, it is necessary to extend covariate for reducing the variance of error components of layers. Suppose we extend the covariate of layers given data (\mathbf{X}, \mathbf{y}) in the following way.

$$\begin{aligned} \mathbf{X}^{(0)} &= \mathbf{X} \\ \mathbf{X}^{(\ell)} &= [\mathbf{X}^{(0)} : \mathbf{x}_e^{(\ell-1)}] \quad \text{for } \ell = 1, \dots, L \\ \mathbf{x}_e^{(\ell)} &= \frac{c_2}{\pi} \{\arctan(\mathbf{d}^{(\ell)} - c_1) + \arctan(\mathbf{d}^{(\ell)} + c_1)\} \quad \text{for } \ell = 1, \dots, L \\ \mathbf{d}^{(\ell)} &= [d_1^{(\ell)}, \dots, d_n^{(\ell)}]^T \quad \text{for } \ell = 1, \dots, L \\ d_i^{(0)} &= \hat{y}'_i - \hat{y}_i \quad \text{for } i = 1, \dots, n \\ d_i^{(\ell)} &= \hat{e}'_{\ell-1,i} - \hat{e}_{\ell-1,i} \quad \text{for } \ell = 1, \dots, L \text{ and } i = 1, \dots, n \end{aligned} \quad (3.3.2)$$

where \hat{y}'_i and \hat{y}_i are estimates of target y_i ; $\hat{e}'_{\ell,i}$ and $\hat{e}_{\ell,i}$ are estimates of target value $e_{\ell,i}$ by oDPLM and DPLM, respectively; c_1 and c_2 are non-zero positive constants; ℓ and i are indices for layer and observation, respectively.

Assuming $|e_{\ell,i} - \hat{e}'_{\ell,i}| \leq |e_{\ell,i} - \hat{e}_{\ell,i}|$, that is, $\hat{e}'_{\ell,i}$ is more close to $e_{\ell,i}$ than $\hat{e}_{\ell,i}$, the covariate extension process is equivalent to the generation of random effect variables categorized into three cases such as over-, under- or well-fitting of targets in the DPLM framework. Define $S^- = \{e_{\ell,i} | d_i^{(\ell)} < -c_1, i = 1, \dots, n\}$, $S^+ = \{e_{\ell,i} | d_i^{(\ell)} \geq c_1, i = 1, \dots, n\}$ and $S^0 = \{e_{\ell,i} | -c_1 \leq d_i^{(\ell)} < c_1, i = 1, \dots, n\}$. Let $n_n = |S^-|$, $n_p = |S^+|$, $n_0 = n - n_n - n_p$. Without loss of generality, we can assume that $E(S^+) = a \geq 0$. Since DPLM assumes that random error of a class is *i.i.d.* normal centered to 0 and random errors of classes are independent, the expected value of e_ℓ will

be 0 (i.e., $E(e_\ell) = 0$). Therefore, by symmetry, $E(S^-)$ and $E(S^0)$ will be $-a$ and 0, respectively (i.e., $E(S^-) = -a$ and $E(S^0) = 0$). Also, due to the independence assumption of DPLM, the variance component of layer ℓ can be decomposed as follows.

$$\begin{aligned}
\text{var}(e_\ell) &= \frac{1}{n} \sum_{i=1}^n (e_{\ell,i})^2 \\
&= \frac{1}{n} \left(\sum_{e \in S^+} e^2 + \sum_{e \in S^-} e^2 + \sum_{e \in S^0} e^2 \right) \\
&= \frac{1}{n} \left\{ \sum_{e \in S^+} (e - a + a)^2 + \sum_{e \in S^-} (e + a - a)^2 + \sum_{e \in S^0} e^2 \right\} \\
&= \frac{n_p}{n} \text{var}(S^+) + \frac{n_n}{n} \text{var}(S^-) + \frac{n_0}{n} \text{var}(S^0) + \frac{n_p + n_n}{n} a^2 \tag{3.3.3}
\end{aligned}$$

The variance component of layer $(\ell + 1)$ can also be written by

$$\begin{aligned}
\text{var}(e_{\ell+1}) &= \text{var}(e_\ell | \mathbf{X}^{(\ell+1)}) \\
&= \text{var}(e_\ell | \mathbf{X}^{(\ell+1)}, \mathbf{x}_e^{(\ell)} \geq \frac{c_2}{2}) p(\mathbf{x}_e^{(\ell)} \geq \frac{c_2}{2} | \mathbf{X}^{(\ell+1)}) \\
&\quad + \text{var}(e_\ell | \mathbf{X}^{(\ell+1)}, \mathbf{x}_e^{(\ell)} < \frac{-c_2}{2}) p(\mathbf{x}_e^{(\ell)} < \frac{-c_2}{2} | \mathbf{X}^{(\ell+1)}) \\
&\quad + \text{var}(e_\ell | \mathbf{X}^{(\ell+1)}, \frac{-c_2}{2} \leq \mathbf{x}_e^{(\ell)} < \frac{c_2}{2}) p(\frac{-c_2}{2} \leq \mathbf{x}_e^{(\ell)} < \frac{c_2}{2} | \mathbf{X}^{(\ell+1)}) \\
&= \frac{n_p}{n} \text{var}(S^+) + \frac{n_n}{n} \text{var}(S^-) + \frac{n_0}{n} \text{var}(S^0). \tag{3.3.4}
\end{aligned}$$

Thus, the covariate extension, described by Eq. 3.3.2, reduces variance by $\frac{n_p + n_n}{n} a^2$ in the next layer. If error is well distributed, then we can assume that S^0 has a normal distribution $\mathcal{N}(0, \sigma)$, both S^- and S^+ have truncated normal distributions of $\mathcal{N}(-c_2, \sigma)$ and $\mathcal{N}(c_2, \sigma)$ in $[-c_2, \infty)$ and $(-\infty, c_2]$, respectively, and $n_p = n_n = n_0$ (due to arc-tangent function). According to *three-sigma rule* [31] stating that nearly all values lie within 3 standard deviations

from mean, the standard deviation σ of S^0 is $\frac{c_2}{6}$,

$$\begin{aligned}
a &= E(S^+) \\
&= c_2 - E(S^0 | S^0 > 0) \\
&= c_2 - \frac{c_2}{6} \frac{\phi(0)}{1 - \Phi(0)}
\end{aligned} \tag{3.3.5}$$

and

$$\begin{aligned}
\text{var}(S^+) &= \text{var}(S^-) \\
&= \text{var}(S^0 | S^0 > 0) \\
&= \left(\frac{c_2}{6}\right)^2 \left\{ 1 - \left(\frac{\phi(0)}{1 - \Phi(0)}\right)^2 \right\}
\end{aligned} \tag{3.3.6}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density and cumulative distribution functions of the standard normal distribution, respectively. In result, when error is well distributed, the rate of variance reduction is

$$\frac{\text{var}(e_t)}{\text{var}(e_{t+1})} = \frac{\left(\frac{c_2}{6}\right)^2 + 2\left(\frac{c_2}{6}\right)^2 \left\{ 1 - \left(\frac{\phi(0)}{1 - \Phi(0)}\right)^2 \right\} + 2\left\{ c_2 - \frac{c_2}{6} \frac{\phi(0)}{1 - \Phi(0)} \right\}}{\left(\frac{c_2}{6}\right)^2 + 2\left(\frac{c_2}{6}\right)^2 \left\{ 1 - \left(\frac{\phi(0)}{1 - \Phi(0)}\right)^2 \right\}} \approx 1.737. \tag{3.3.7}$$

3.4 Overfit Models of Layers

It is no wonder that we can always find an overfit model given a performance measurement. For instance, including more explanatory variables to a regression model will produce a model which reduces the mean square error over a dataset. In the DPLM framework, an overfit model can be obtained by adding designed noises during parameter sampling.

In general, Eq. 2.5.41 cannot be computed analytically because of non-closed-form

integrals involved. However, it can be computed approximately using the Monte Carlo integration. Suppose that $\theta^{(t)}$ is the set of model parameters sampled at iteration t counted after burn-in period for $t = 1, \dots, T$. Given the value of x , the response estimate will be approximately

$$E(y|x) \approx \frac{1}{T} \sum_{t=1}^T E(y|x, \theta^{(t)}) \quad (3.4.1)$$

and the model-wise estimate will be

$$E(y|x, \theta^{(t)}) = \frac{\sum_{j=1}^K E(y|x, \theta_j^{(t)}) f_x(x|\theta_j^{(t)}) + \alpha^{(t)} \int E(y|x, \theta) f_x(x|\theta) G_0(d\theta)}{\sum_{j=1}^K f_x(x|\theta_j^{(t)}) + \alpha^{(t)} \int f_x(x|\theta) G_0(d\theta)}. \quad (3.4.2)$$

During the parameter sampling at some iteration $s \in [1, T]$, we add a new component ($K+1$) such that $E(y|x, \theta_{K+1}^{(s)}) = y_i$ and $f_x(x_i|\theta_{K+1}^{(s)}) \gg f_x(x_i|\theta_j^{(s)}), \forall j \in [1, K]$, where (x_i, y_i) is a covariance-response pair in the training dataset. Assuming that $\alpha^{(s)}$ is neglectable (which is true in most cases after burn-in), the model-wise estimate at iteration s is

$$E(y|x_i, \theta^{(s)}) \approx y_i. \quad (3.4.3)$$

Suppose that there are two DPLM models, denoted by DPLM and oDPLM separately and that we add such a new component as a designed noise during the parameter sampling of oDPLM. Let S be the set of iterations at which such a new component is added for a covariate-response pair (x_i, y_i) . Also, let $\hat{y}_{i,t}$ and $\hat{y}'_{i,t}$ be the model-wise response estimates for y_i at iteration t by DPLM and oDPLM, respectively. Then the ratio of mean square errors

of oDPLM over DPML for (x_i, y_i) is

$$\frac{\sum_{t=1}^T (\hat{y}'_{i,t} - y_i)^2}{\sum_{t=1}^T (\hat{y}_{i,t} - y_i)^2} = \frac{\sum_{t \notin S} (\hat{y}'_{i,t} - y_i)^2}{\sum_{t \notin S} (\hat{y}_{i,t} - y_i)^2 + \sum_{t \in S} (\hat{y}_{i,t} - y_i)^2}. \quad (3.4.4)$$

Assuming $\sum_{t \notin S} (\hat{y}'_{i,t} - y_i)^2 \approx \sum_{t \notin S} (\hat{y}_{i,t} - y_i)^2$, which is achievable by setting $|S| \ll |T|$,

$$\sum_{t=1}^T (\hat{y}'_{i,t} - y_i)^2 \leq \sum_{t=1}^T (\hat{y}_{i,t} - y_i)^2. \quad (3.4.5)$$

Similarly, for covariate-response pairs (\mathbf{X}, \mathbf{Y}) , an overfit DPLM in the layer ℓ of MDPLM, denoted by oDPML $_{\ell}$, can be also constructed by adding designed noises during parameter sampling.

3.5 The Layer of Prediction

The variance reduction of MDPLM was proved in Section 3.3, but the reduction of variance in a higher layer does not mean that the prediction at that layer will be better than at lower layers because of data overfitting. Since MDPLM utilizes model overfitting to find unobservable random effects, this may cause MDPLM to overfit a training dataset if the found random effects will be valid only in the training dataset. Thus, choosing a proper layer of prediction would result in a robust MDPLM avoiding overfitting.

3.5.1 Performance Measure

The best model balances training (observed) and validating (unobserved) errors. Since it is common that the size of training data is larger than that of validation, the simple arithmetic mean of two measurements tends to be biased and the training data dominates

the performance measure. Therefore, we use a different way of merging two mean square errors of training and validation sets as our performance measure to resolve uncertainty when two measurements are very different and call it $cMSE$ (defined in Eq. 3.5.1).

$$cMSE = p \cdot MSE_t + (1 - p) \cdot MSE_v \quad (3.5.1)$$

$$v_1 = 1 - \frac{MSE_t}{MSE_t + MSE_v} \quad \text{and} \quad v_2 = 1 - v_1$$

$$h_1 = -v_1 \log_2 v_1 \quad \text{and} \quad h_2 = -v_2 \log_2 v_2$$

$$p = \frac{h_1}{h_1 + h_2}$$

where MSE_t and MSE_v are mean square errors of training and validation, respectively.

The value of $cMSE$ ranges between MSE_t and MSE_v , inclusive, and gets closer to the larger of two. Hence, it is useful to check robustness to unseen (or validation) data while considering fitness to training data.

3.5.2 Optimal Path to the Layer of Prediction

There are two kinds of noises involved in measuring the performance of an MDPLM. The first is the designed noise to make overfit models of layers. The rate (or frequency) of designed noises, during parameter sampling, determines the level of deviation in overfit estimates from estimates of a layer. As the rate gets higher, the estimates by oDPLM approach target values closer and departs further from estimates by DPLM in that layer. That is, the rate of designed noises decides the level of overfitting (actually, things are more complicated than this because of unpredictable interactions, which will be explained later). The other noise is the decision of layer where the final prediction is retained. The term *noise* is used in the

sense that a layer transition incurs a cumulative noise effect resulting from the history of random effect generations from the first layer. The variance reduction of MDPLM by layer for training dataset was proved in Section 3.3, but there is a point where a generated random effect by our covariate extension fails to be a common characteristic of both training and validation datasets. In other words, the generated random effect does not contribute to the reduction of MSE_v and results in the increase of $cMSE$ after some layer. In fact those two kinds of noises are interrelated because the decision of noise rate in a layer affects the next layer. Therefore the designed noise rate and the layer of prediction make a path to an optimal model minimizing the performance measure $cMSE$.

Let $cMSE(\eta, \ell)$ be the performance measure of MDPLM where the layer of prediction is ℓ and the frequency path (or sequence) of designed noise rates from layer 0 to layer $(\ell - 1)$ is $\eta = \{\eta_0, \dots, \eta_{\ell-1}\}$. The rate η_i denotes the proportion of iterations at layer i , when the new components (described in Section 3.4) are added, over the total number of iterations after burn-in period during parameter sampling. When more than one training-validation dataset pairs are considered, an optimal model minimizing $E(cMSE)$ can be obtained by

$$\min_{(\eta, \ell)} E[cMSE(\eta, \ell)]. \quad (3.5.2)$$

Unfortunately, the problem is not solvable in a feasible amount of time because $E(cMSE)$ cannot be expressed analytically in terms of η and L . We will construct a multi-stage decision problem for the layer of prediction in Section 3.5.3 and discuss an efficient way in the MDPLM framework in Section 3.5.5.

3.5.3 Multi-stage Decision Problem for the Layer of Prediction

Consider a situation to determine the layer, between 0 and 1, minimizing our performance measure $cMSE$ for a training-validation dataset pair $(\mathbf{D}_t, \mathbf{D}_v)$ where $\mathbf{D}_t = (\mathbf{X}_t, \mathbf{y}_t)$ and $\mathbf{D}_v = (\mathbf{X}_v, \mathbf{y}_v)$. For the time being, we assume that parameters are only regression coefficients and that errors of a layer are *i.i.d.* throughout mixture components for explanation purpose. Suppose that $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ are two vectors representing regression coefficients of layer 0 and 1, respectively and that $\mathbf{z}_t^{(0)}$ and $\mathbf{z}_v^{(0)}$ are random effect vectors generated for training and validation datasets, respectively. The layer of prediction can be determined by solving the following problem with a decision parameter η_0 which represents the designed noise rate used to make an overfit model (see Section 3.4).

$$\begin{aligned} & \min_{\eta_0} cMSE(\mathbf{D}_t, \mathbf{D}_v) \\ MSE_t &= \frac{1}{N_t} \left| \mathbf{y}_t - \mathbf{X}_t \mathbf{c}^{(0)} - [\mathbf{X}_t : \mathbf{z}_t^{(0)}] \mathbf{c}^{(1)} \right|_2^2 \\ MSE_v &= \frac{1}{N_v} \left| \mathbf{y}_v - \mathbf{X}_v \mathbf{c}^{(0)} - [\mathbf{X}_v : \mathbf{z}_v^{(0)}] \mathbf{c}^{(1)} \right|_2^2 \end{aligned} \quad (3.5.3)$$

where N_t and N_v are the sizes of training and validation sets, respectively; $|\cdot|_2$ denotes L_2 -norm.

Suppose that the random effect vectors $\mathbf{z}_t^{(0)}$ and $\mathbf{z}_v^{(0)}$ are generated in the way described in Section 3.3 (we use \mathbf{z} for generated random effects here rather than \mathbf{x}_e). Then, the regression coefficient $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ are obtained by solving a two-stage decision problem because the decision of $\mathbf{c}^{(1)}$ requires the value of $\mathbf{c}^{(0)}$.

$$\text{Stage one: } \min_{\mathbf{c}^{(0)}} \left| \mathbf{e}^{(0)} \right|_2 \quad \text{s.t. } \mathbf{e}^{(0)} = \mathbf{y}_t - \mathbf{X}_t \mathbf{c}^{(0)} \quad (3.5.4)$$

$$\text{Stage two: } \min_{\mathbf{c}^{(1)}} \left| \mathbf{e}^{(0)} - [\mathbf{X}_t : \mathbf{z}_t^{(0)}] \mathbf{c}^{(1)} \right|_2 \quad (3.5.5)$$

Actually, MDPLM does not estimate parameters, that is, it computes $\mathbf{X}_t \mathbf{c}^{(0)}$ and $[\mathbf{X}_t : \mathbf{z}_t^{(0)}] \mathbf{c}^{(1)}$ directly from sampled parameters, but the computation is also sequentially executed as the above two-stage problem. When $\eta_0 = 0$, that is, $cMSE$ is minimized at layer 0, the minimized value of Eq. 3.5.5 will be the same as the minimized value of Eq. 3.5.4 because $\mathbf{z}_t^{(0)} = \mathbf{0}$ (see Eq. 3.3.1). In case of MDPLM, the result is also preserved because the response estimate vectors computed by $[\mathbf{X}_t : \mathbf{z}_t^{(0)}] \mathbf{c}^{(1)}$ will be $\mathbf{0}$.

Now let us extend the approach upto layer $L (\geq 2)$. Let a vector $\mathbf{c}^{(\ell)}$ be the regression coefficient of layer ℓ . Also, let $\mathbf{z}_t^{(\ell-1)}$ and $\mathbf{z}_v^{(\ell-1)}$ be the random effect vectors generated at layer ℓ for training and validation datasets, respectively. The formula corresponding to Eq. 3.5.3 will be the same except the part of response estimation, which is the sum of estimates by layers.

$$\begin{aligned} & \min_{\eta_{0:(L-1)}} cMSE(\mathbf{D}_t, \mathbf{D}_v) \\ MSE_t &= \frac{1}{N_t} \left| \mathbf{y}_t - \mathbf{X}_t \mathbf{c}^{(0)} - \sum_{\ell=1}^L [\mathbf{X}_t : \mathbf{z}_t^{(\ell-1)}] \mathbf{c}^{(\ell)} \right|_2^2 \\ MSE_v &= \frac{1}{N_v} \left| \mathbf{y}_v - \mathbf{X}_v \mathbf{c}^{(0)} - \sum_{\ell=1}^L [\mathbf{X}_v : \mathbf{z}_v^{(\ell-1)}] \mathbf{c}^{(\ell)} \right|_2^2 \end{aligned} \quad (3.5.6)$$

The two-stage problem to compute the regression coefficient vectors of layers are also extended to a multi-stage problem shown in Eq. 3.5.7 and Eq. 3.5.8 because the generation of random effect vector at a layer requires the coefficient vector (or response estimates in

MDPLM) of the previous layer.

$$\text{Stage one: } \min_{\mathbf{c}^{(0)}} \left| \mathbf{e}^{(0)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}^{(0)} = \mathbf{y}_t - \mathbf{X}_t \mathbf{c}^{(0)} \quad (3.5.7)$$

$$\text{Stage } \ell: \min_{\mathbf{c}^{(\ell)}} \left| \mathbf{e}^{(\ell)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}^{(\ell)} = \mathbf{e}^{(\ell-1)} - [\mathbf{X}_t : \mathbf{z}_t^{(\ell-1)}] \mathbf{c}^{(\ell)} \quad , \ell = 2, \dots, L-1 \quad (3.5.8)$$

For a less biased performance measure, we extend Eq. 3.5.6 by minimizing the expected value of $cMSE$ over more than one training-validation dataset pair. Let M be the number of training-validation dataset pairs, that is, we extract, from the entire dataset \mathbf{D} , M dataset pairs $(\mathbf{D}_{t_m}, \mathbf{D}_{v_m})_{m=1}^M$. Assuming that such pairs have the same probability to occur and letting the index m denote a dataset pair to which matrices or vectors of the index are related,

$$\begin{aligned} \min_{\eta_{0:(L-1)}} E [cMSE(\mathbf{D})] &= \frac{1}{M} \sum_{m=1}^M cMSE(\mathbf{D}_{t_m}, \mathbf{D}_{v_m}) \\ MSE_{t_m} &= \frac{1}{N_t} \left| \mathbf{y}_{t_m} - \mathbf{X}_{t_m} \mathbf{c}_m^{(0)} - \sum_{\ell=1}^L [\mathbf{X}_{t_m} : \mathbf{z}_{t_m}^{(\ell-1)}] \mathbf{c}_m^{(\ell)} \right|_2^2, \quad m = 1, \dots, M \\ MSE_{v_m} &= \frac{1}{N_v} \left| \mathbf{y}_{v_m} - \mathbf{X}_{v_m} \mathbf{c}_m^{(0)} - \sum_{\ell=1}^L [\mathbf{X}_{v_m} : \mathbf{z}_{v_m}^{(\ell-1)}] \mathbf{c}_m^{(\ell)} \right|_2^2, \quad m = 1, \dots, M \end{aligned} \quad (3.5.9)$$

The extension to the multi-stage problem of Eq. 3.5.8 has the same form, but the layers of the same index in different dataset pairs are interrelated in that the layers share the same noise rate η_ℓ . Since the computation of regression coefficient vector (or the response estimate vector in MDPLM) in a layer requires those of previous layers in all datasets, the extension

still forms a multi-stage problem as described in Eq. 3.5.10 and Eq. 3.5.11.

$$\text{Stage one: } \min_{\mathbf{c}_m^{(0)}} \left| \mathbf{e}_m^{(0)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}_m^{(0)} = \mathbf{y}_{\mathbf{t}m} - \mathbf{X}_{\mathbf{t}m} \mathbf{c}_m^{(0)} \quad , m = 1, \dots, M \quad (3.5.10)$$

$$\text{Stage } \ell: \min_{\mathbf{c}_m^{(\ell)}} \left| \mathbf{e}_m^{(\ell)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}_m^{(\ell)} = \mathbf{e}_m^{(\ell-1)} - [\mathbf{X}_{\mathbf{t}m} : \mathbf{z}_{\mathbf{t}m}^{(\ell-1)}] \mathbf{c}_m^{(\ell)} \quad , m = 1, \dots, M \quad (3.5.11)$$

$$, \ell = 2, \dots, L-1$$

Let us revert to the MDPL framework; $\theta^{(\ell)}$ denotes the set of parameters in the DPLM at layer ℓ ; $\hat{\mathbf{y}}$ (or $\hat{\mathbf{e}}$) is a vector of estimates for \mathbf{y} (or \mathbf{e}) of corresponding index; the regression problem for a training dataset $\mathbf{D}_{\mathbf{t}m}$ at state ℓ has $\mathbf{X}_{\mathbf{t}m}^{(\ell-1)}$ as its covariate. Then the multi-stage decision problem can be written as follows.

$$\min_{\eta_{0:(L-1)}} E [cMSE(\mathbf{D})] = \frac{1}{M} \sum_{m=1}^M cMSE(\mathbf{D}_{\mathbf{t}m}, \mathbf{D}_{\mathbf{v}m})$$

$$MSE_{\mathbf{t}m} = \frac{1}{N_{\mathbf{t}}} \left| \mathbf{y}_{\mathbf{t}m} - \hat{\mathbf{y}}_{\mathbf{t}m} \right|_2^2 \quad , m = 1, \dots, M \quad (3.5.12)$$

$$MSE_{\mathbf{v}m} = \frac{1}{N_{\mathbf{v}}} \left| \mathbf{y}_{\mathbf{v}m} - \hat{\mathbf{y}}_{\mathbf{v}m} \right|_2^2 \quad , m = 1, \dots, M$$

$$\text{Stage one: } \min_{\theta^{(0)}} \left| \mathbf{e}_m^{(0)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}_m^{(0)} = \mathbf{y}_{\mathbf{t}m} - \hat{\mathbf{y}}_{\mathbf{t}m} \quad , m = 1, \dots, M \quad (3.5.13)$$

$$\text{Stage } \ell: \min_{\theta^{(\ell)}} \left| \mathbf{e}_m^{(\ell)} \right|_2 \quad \text{s.t.} \quad \mathbf{e}_m^{(\ell)} = \mathbf{e}_m^{(\ell-1)} - \hat{\mathbf{e}}_m^{(\ell-1)} \quad , m = 1, \dots, M \quad (3.5.14)$$

$$, \ell = 2, \dots, L-1$$

Finding an optimal sequence of designed noise rates is computationally prohibitive. The performance measure $E[cMSE]$ has no analytical expression. The decision of noise rate at a layer generates the different response distribution and random effects in the next layer.

Suppose that we discretize the designed noise η_ℓ so that η_ℓ can take on one of d values and that there are M training-validation dataset pairs. If we enumerate all the possible sequences of designed noise rates to layer L , there is a total of d^L scenarios. Furthermore, except the last layer, every layer of MDPLM consists of two DPLMs (DPLM and oDPLM). Therefore, if we go through all the possible paths, we will have to solve $2M \left(\frac{1-d^{L+1}}{1-d} - \frac{1}{2} \right)$ DPLMs. In Section 3.5.5, we will discuss how to solve the multi-stage decision problem efficiently in our frame.

3.5.4 Generating training-validation dataset pairs

Heterogeneous data are filled with lots of uncertainties and it is difficult to obtain a stratified training-validation dataset pair. Using a DPLM trained with the whole data $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$, the expected number of mixture components can be computed by $E(K) = \frac{1}{T} \sum_{t=1}^T K^{(t)}$ where $K^{(t)}$ is the number of mixture components at iteration t and T is an arbitrary large number which denotes the number of iterations after burn-in period. Under the assumption that each class takes the same number of data, in order to make a training set have approximate half of instances for each mixture component, the size of validation sets is set to

$$\frac{n}{2E(K)} \tag{3.5.15}$$

where n is the number of observations in \mathbf{D} .

For a less biased performance measure, M -fold cross-validation is used to compute $E(cMSE)$ with modification. We sample M validation sets, denoted by \mathbf{D}_{v_m} for $m = 1, \dots, M$, without replacement and the training sets, denoted by \mathbf{D}_{t_m} for $m = 1, \dots, M$ such that $\mathbf{D}_{t_m} = \mathbf{D} \setminus \mathbf{D}_{v_m}$. Let $cMSE_m$ be the performance measure of $(\mathbf{D}_{t_m}, \mathbf{D}_{v_m})$. Then the comprehensive performance measure is approximately $E(cMSE) \approx \frac{1}{M} \sum_{m=1}^M cMSE_m$ as mentioned before.

3.5.5 An Alternative Way for the Layer of Prediction

Solving the multi-stage problem in Eq. 3.5.12, Eq. 3.5.13 and Eq. 3.5.13 for an optimal layer of prediction is not possible in a feasible amount of time in most cases. Alternatively, we take advantage of the properties of the designed noise introduced during sampling. For better understanding how the designed noise affects an overfit model, it is needed to know the detailed implementation of introducing noise to sampling. Let η_ℓ be the rate of designed noise resulting in the overfit model at layer ℓ , oDPLM_ℓ , and N_{iter} be the number of iterations after burn-in period. The noise rate η_ℓ has a value between 0 and 1, inclusive, and this implies that, every $1/\eta_\ell$ iterations in average, one of data observations in the training dataset is selected randomly, uniformly over N_t , the size of training dataset, and a new component, centered on the selected observation, is added. Therefore, the probability that an observation in the training dataset forms a new centered component at an iteration is η_ℓ/N_t . In spite of such a low chance, it is very unlikely to occur that a component will not participate in the creation of centered components during sampling since N_{iter} is much larger than N_t ($N_t \ll N_{iter}$).

The primary role of vector $\mathbf{d}^{(\ell)}$, in our covariate extension of Eq. 3.3.2, is to estimate the direction of errors. The arc-tangent function forces over- or under-estimated values to follow a truncated normal distribution. Therefore, if a centered component did not affect other estimates but an involved target, it would be expected that MSE_t (or $E(MSE_t)$) is decreasing and asymptotically converging to 0 over error rate. However this should be avoided for robustness to unseen data and, therefore, the precision matrix s_x of the centered component is sampled from the base distribution G_0 of Dirichlet process to cover a wide covariate region. In detail, when x_i is an observation to form a centered mixture component of random

covariate vector X_{K+1} , X_{K+1} is assumed to have the distribution

$$X_{K+1}|x_i, s_{x,(K+1)} \sim \mathcal{N}(x_i, s_{x,(K+1)}^{-1}) \quad (\text{i.e., } \mu_{K+1} = x_i) \quad (3.5.16)$$

and the precision parameter $s_{x,(K+1)}$ is sampled from

$$s_{x,(K+1)} \sim \mathcal{W}(\rho, \frac{1}{\rho} w^{-1}) \quad (3.5.17)$$

(see Figure 2.5.2 and Table 2.5.3 for parameters μ , ρ and w).

Also, the regression coefficient vector β_{K+1} is set to an *ordinary least square* (OLS) solution so that the component-wise estimate should be y_i .

For such a centered component, the overfit property (Eq. 3.4.5) is still preserved. Let $\hat{y}'_{i,t}$ and $\hat{y}_{i,t}$ be model-wise estimates for y_i with and without a centered component, respectively, at iteration t . Also, let $y_{i,t,j}$ denote a component-wise estimate for y_i in the component j of the model at iteration t . For the sake of simplicity, we rewrite Eq. 3.4.2 by ignoring the effect of α , and define g' and g as the normalized probability functions for f_x with and without a concentrated component, respectively.

Then

$$\begin{aligned} g_j(x) &= \frac{f_x(x|\theta_j^{(t)})}{\sum_{k=1}^K f_x(x|\theta_k^{(t)})} & \text{for } j=1, \dots, K \\ g'_j(x) &= \frac{f_x(x|\theta_j^{(t)})}{\sum_{k=1}^{K+1} f_x(x|\theta_k^{(t)})} & \text{for } j=1, \dots, K+1 \end{aligned} \quad (3.5.18)$$

and

$$\hat{y}_{i,t} = \sum_{j=1}^K y_{i,t,j} g_j(x_i) \quad (3.5.19)$$

$$\hat{y}'_{i,t} = \sum_{j=1}^{K+1} y_{i,t,j} g'_j(x_i). \quad (3.5.20)$$

Since the estimate for y_i from the concentrate component is y_i (i.e., $y_{i,t,K+1} = E(y|x_i, \theta_{K+1}^{(t)}) = y_i$), the relation of two estimates is given by

$$\begin{aligned} \hat{y}'_{i,t} &= \sum_{j=1}^K y_{i,t,j} \frac{f_x(x|\theta_j^{(t)})}{\sum_{k=1}^K f_x(x|\theta_k^{(t)})} \cdot \frac{\sum_{k=1}^K f_x(x|\theta_k^{(t)})}{\sum_{k=1}^{K+1} f_x(x|\theta_k^{(t)})} + y_i \cdot \frac{f_x(x|\theta_{K+1}^{(t)})}{\sum_{k=1}^{K+1} f_x(x|\theta_k^{(t)})} \\ &= \hat{y}_{i,t} \left(1 - g'_{K+1}(x_i)\right) + y_i g'_{K+1}(x_i). \end{aligned} \quad (3.5.21)$$

Then the difference of squared estimation errors from two models at iteration t is

$$\begin{aligned} (y_i - \hat{y}_{i,t})^2 - (y_i - \hat{y}'_{i,t})^2 &= (y_i - \hat{y}_{i,t})^2 - \left\{ y_i - \hat{y}_{i,t} \left(1 - g'_{K+1}(x_i)\right) + y_i g'_{K+1}(x_i) \right\}^2 \\ &= (y_i - \hat{y}_{i,t})^2 + \left(1 - g'_{K+1}(x_i)\right)^2 (y_i - \hat{y}_{i,t})^2 \\ &= g'_{K+1}(x_i) \left(2 - g'_{K+1}(x_i)\right) (y_i - \hat{y}_{i,t})^2 \geq 0. \end{aligned} \quad (3.5.22)$$

When S is the set of iterations at which a new component is added for (x_i, y_i) , the ratio of mean square error of oDPLM over DPLM for (x_i, y_i) is

$$\begin{aligned} \frac{\sum_{t=1}^T (\hat{y}'_{i,t} - y_i)^2}{\sum_{t=1}^T (\hat{y}_{i,t} - y_i)^2} &= \frac{\sum_{t \notin S} (\hat{y}'_{i,t} - y_i)^2 + \sum_{t \in S} (\hat{y}'_{i,t} - y_i)^2}{\sum_{t \notin S} (\hat{y}_{i,t} - y_i)^2 + \sum_{t \in S} (\hat{y}_{i,t} - y_i)^2} \\ &\leq 1 \end{aligned} \quad (3.5.23)$$

and the overfit property for (x_i, y_i) is preserved.

Sampling the precision matrix $s_{x,(K+1)}$ from G_0 will produce a flat and wide covariate region for the component and result in the increase of robustness to unseen (or validation) data. However, it will make a centered component affect the estimates of other observations and increase complexity in the relationship between noise rate and error measures. Figure 3.5.1 depicts error measures $E(MSE_t)$, $E(MSE_v)$ and $E(cMSE)$ over designed noise rate η_0 at layer 1 for the data distribution described in Eq. 4.1.1. Our performance measure $E(cMSE)$ has low values over the entire range of noise rate except some peculiar points represented by high peaks or little bumps. Those peculiar points are more clearly visible when we plot the sorted $E(cMSE)$ in Figure 3.5.2, where $E(cMSE)$ is relative high at only about 10% of the designed noise rates considered. One of the reasons for high $E(cMSE)$ is that centered components works adversary to other estimates by interaction. Such high values are also observed in $E(MSE_t)$ at $\eta_0 = 0.03, 0.10, 0.21, 0.31, 0.47, 0.52$ in the first plot of Figure 3.5.1. The other reason is that the random effect generated by designed noise introduction loses robustness to validation data. The relative high values of $E(MSE_v)$ at $\eta_0 = 0.03, 0.21, 0.52, 0.64, 0.67, 0.89$ exemplify this situation in the second plot of Figure 3.5.1

Assuming that error measures follow such observations, that is, $E(cMSE)$ is low over the entire range of designed noise rate except at some peculiar points, and that lower designed noise rate is preferred for robustness issue, we can reduce the search space of designed noise rate while obtaining a satisfiable performance measure value close to the minimum in a layer. Of course, such reduction is dependent on data distributions of both training and validation datasets and ,therefore, it is impossible to obtain a generalized formula, but the designed noise rate range $[0,0.1]$ of 10 grid is enough to produce a satisfiable performance measure close to the minimum achieved by searching over the range $[0,1]$ of 100 grid in most cases and this result is experimentally obtained in simulated datasets.

MDPLM can be considered as a nonlinear system where a dataset (\mathbf{X}, \mathbf{Y}) is input, the

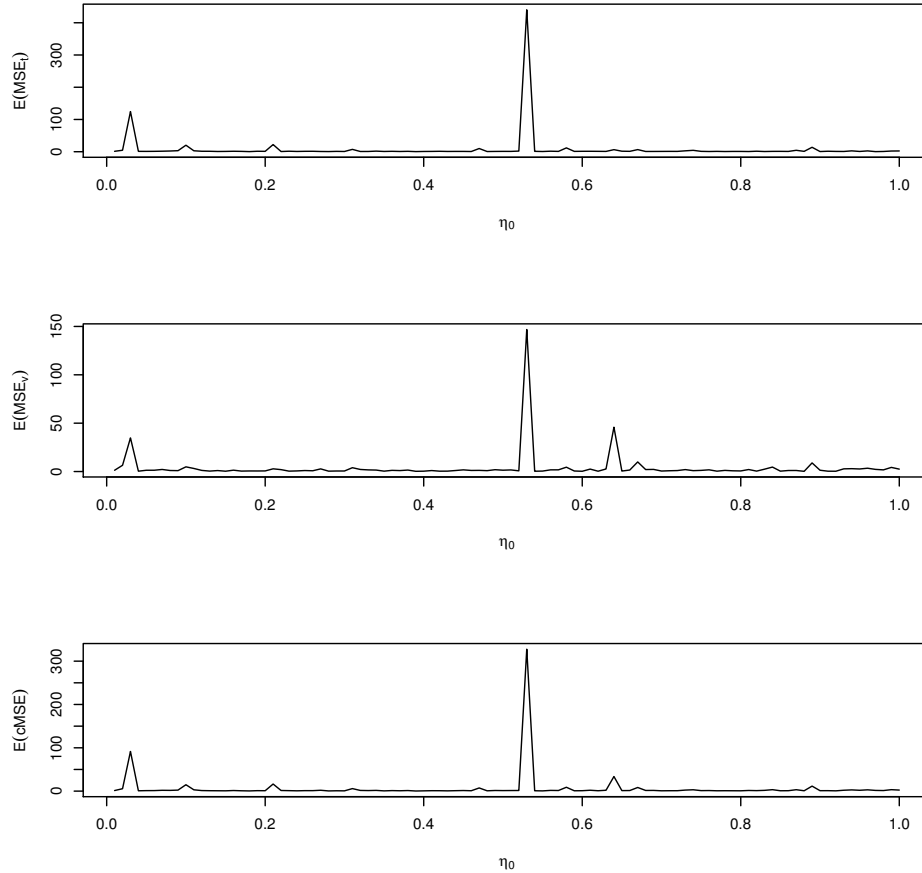


Figure 3.5.1: The plots of designed noise rate η_0 vs error measures ($E(MSE_t)$, $E(MSE_v)$ and $E(cMSE)$) at layer 1 for data distribution described in Eq. 4.1.1: performance measure $E(cMSE)$ depends on the distribution of data observations which forms centered components

performance measure is output, and the random effect variables, generated in layers, are noise introduced. In the stochastic resonance ([32, 33]) framework where there is an optimal level of noise to minimize (or maximize) a performance measure, we can regard the increase of layers in MDPLM as the level of noise in the MDPLM system because more layers involve more random effects generated. Such a system-wise representation of the MDPLM is shown in Figure 3.5.3.

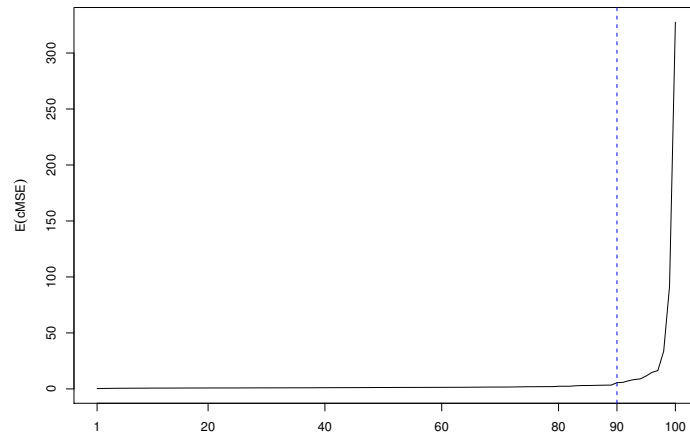


Figure 3.5.2: The plots of $E(cMSE)$ sorted in increasing order: about 90% of designed noise rates produce low performance measure.

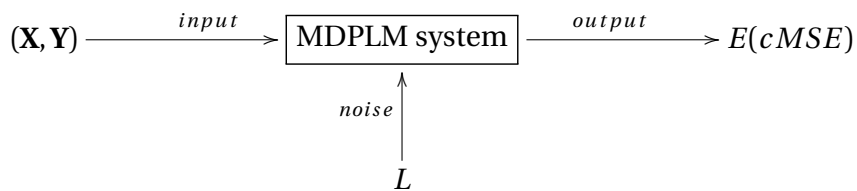


Figure 3.5.3: The MDPLM system: the system has the covariate-response pairs (\mathbf{X}, \mathbf{Y}) as input, the performance measure $cMSE$ as output, and the layer of prediction L as noise

The difficulty of usual multi-stage problems comes from the fact that it is not guaranteed that a sequence of decisions optimizing the objective function of interest up to a stage is the part of an optimal solution. Even though such a difficulty still exists in finding an optimal path of designed noise rates, its degree can be reduced due to the design of MDPLM. If a model with a certain designed noise sequence lost robustness (that is, high $E(cMSE)$) at a layer, the prediction (or response estimate) by the model would be unstable even though an optimal path had the sequence as a subpath. It is because an extended variable variable x_e

(or hidden random effect) is evaluated at the time of prediction. For instance, suppose that we have an optimal path $\eta^{opt} = (\eta_0^*, \eta_1^*, \eta_2^*)$ with the minimum $E(cMSE)^{opt}$, that is, there are 4 layers (layer 0-3) and layer 3 has a minimum $E(cMSE)^{opt}$. Let $E(cMSE)^{(\ell)}$ be the performance measure at layer ℓ on the path η^{opt} . Suppose that the sequence of performance measures along with layers has the following relationship.

$$E(cMSE)^{opt} = E(cMSE)^{(3)} < E(cMSE)^{(1)} < E(cMSE)^{(2)} < E(cMSE)^{(0)}$$

The above inequality implies that η_1^* produces too-overfitted oDPLM₁ for one or more validation sets and the response estimate from such too-overfit ones has high variance. When an unseen x is given for prediction, an estimated extension variable x_e at layer 1 will be distrustful because of high variance. Thus, the model DPLM₂, which requires x_e as an input, will lose robustness and produce a wrong response estimate. Since the model DPLM₃ also uses the suspicious output of DPLM₂, an estimate from DPLM₃ will also be distrustful even though the performance measure is the lowest. Therefore, the prediction from layer 1 will be safer than from layer 3.

May occur a question that what if $E(cMSE)^{(2)}$ is very close to $E(cMSE)^{(1)}$. In such a case, an extension variable x_e , generated at layer 1, does not participate in regression because either x_e or its regression coefficient approaches zero, which implies that a model has reached its minimum error at layer 1 in the MDPLM framework. Thus, adding more extended variables along with layers will steadily increase variance in prediction and, hence, $E(cMSE)^{(3)}$ will be similar to $E(cMSE)^{(1)}$, that is, $E(cMSE)^{(1)} \approx E(cMSE)^{(2)} \approx E(cMSE)^{(3)}$. Preferring to parsimony, layer 1 will still be the layer of prediction. In result, due to the design of MDPLM and the property of $E(cMSE)$, a sequential search for designed noise rates will produce a robust MDPLM and the procedure is provided in Procedure 1.

Procedure 1 Procedure for the layer of prediction

```
1:  $minCMSE \leftarrow E [cMSE(NIL, 0)]$ 
2:  $\ell \leftarrow 0$ 
3:  $\eta^* \leftarrow NIL$ 
4:  $Cont \leftarrow true$ 
5: while  $Cont$  do
6:    $\eta_\ell^* \leftarrow \underset{0 < \eta_\ell < \eta_{max}}{\operatorname{argmin}} E [cMSE((\eta^*, \eta_\ell), \ell + 1)]$ 
7:    $cMSE \leftarrow E [cMSE((\eta^*, \eta_\ell^*), \ell + 1)]$ 
8:   if  $cMSE < minCMSE$  then
9:      $minCMSE \leftarrow cMSE$ 
10:     $\eta^* \leftarrow (\eta^*, \eta_\ell^*)$ 
11:     $\ell \leftarrow \ell + 1$ 
12:   else
13:      $Cont \leftarrow false$ 
14:   end if
15: end while
16: return  $\ell$  and  $\eta^*$ 
```

Unlike the designed noise rate, the level noise, incurred by layer transition, has a cumulative effect on $E(cMSE)$. While the decision on designed noise rate focuses on the variance reduction of error in a layer conditional on the history of noise rates of previous layers, the layer decision for prediction cares about all possible paths of noise rates to the layer of concern. However, once a sequence of designed noise rates is determined, the layer decision for prediction is trivial due to the same reason for robustness we applied to the decision of designed noise rate. If a path for designed noise rates has a sensitive layer to unseen data, the prediction through the path will be unstable also even though the path achieves a minimum $E(cMSE)$. Therefore a simple greedy search from the first layer will find a robust layer of prediction. Figure 3.5.4 shows the transition of error measures by layers. Since the layer 1 has the minimum $E(cMSE)$ among the layers considered, the layer 1 is selected as the layer of prediction. The actual decision will be made earlier because we do not need to compute $E(cMSE)$ beyond the layer 2.

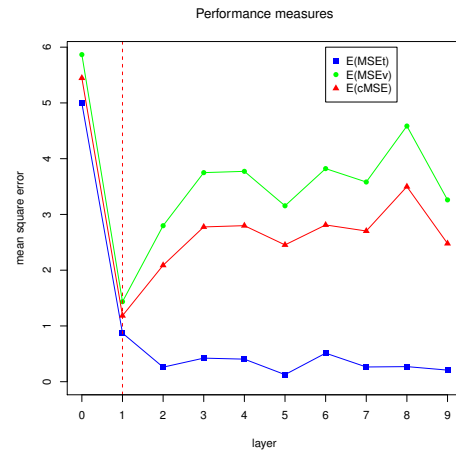


Figure 3.5.4: Transition of error measures by layers (2)

Properties of MDPLM

In this chapter, we will discuss the properties of MDPLM with three simulated datasets. Since mixed models cannot deal with data of unknown number of classes and DPGLM requires the selection of covariate-response and error functions, we compare MDPLM with DPLM in the following properties: hidden random effects, diversity of function complexity, heteroscedasticity.

4.1 Hidden Random effects

When an independent variable (e.g., x or x^2) has a constant effect on the response (e.g., $\frac{\partial y}{\partial x} = c, \forall x$, or $\frac{\partial y}{\partial(x^2)} = c, \forall x^2$, where c is a constant), the variable is said to have a fixed effect (or the variable itself is called a fixed effect). On the contrary, when the effect of a variable (e.g., z) changes over its value (e.g., $y = (c_1x)z + (c_2x)(1 - z)$, $z = \{0, 1\}$), the variable has a

random effect (or the variable itself is called a random effect). Let us call a *hidden random effect** a random effect caused by a hidden variable, which does not exist in data observations. In order to verify the ability of MDPLM in identifying hidden random effects, consider the following data distribution.

$$\begin{aligned}
X &= (x_1, x_2, z) \\
x_1 &\sim \mathcal{U}(0, 1) \\
x_2 &\sim \mathcal{U}(0, 1) \\
z &\sim \mathcal{N}\left(I\left(\frac{x_2}{x_1} > 1\right), \sigma_z\right) \\
y|X &\sim p_1 \mathcal{N}(z, \sigma_z) \mathcal{N}(2x_1 + 4x_2 + 5, \sigma_y) \\
&\quad + p_2 \mathcal{N}(z - 1, \sigma_z) \mathcal{N}(-2x_1 - x_2 + 20, \sigma_y)
\end{aligned} \tag{4.1.1}$$

where $I(c)$ is the indicator function, which is 1 if the condition c is true or 0 otherwise; σ_y , σ_z , p_1 and p_2 are constant.

The covariate X consists of three variables, x_1 , x_2 , and z , and the explanatory variable z has a nonlinear relationship with others. The response y has a mixture relationship with two distinct components; 40 sampled points, denoted by \mathbf{X} , are depicted in Figure 4.1.1. When a DPLM with covariate (x_1, x_2, z) is constructed (see Example 2.5.3 for details), its multivariate normal assumption on covariate makes the distributions of two covariate components clearly separable. The sampled points (from parameter sampling) for covariate mean parameter μ in Figure 4.1.2 show the existence of two distinct components; the distributions of two covariate components are hardly overlapped because two components have separable z

*A hidden random effect in the covariate-extension differs in that it is generated from the difference in estimates between DPLM and oDPLM.

distributions as shown in Figure 4.1.3.

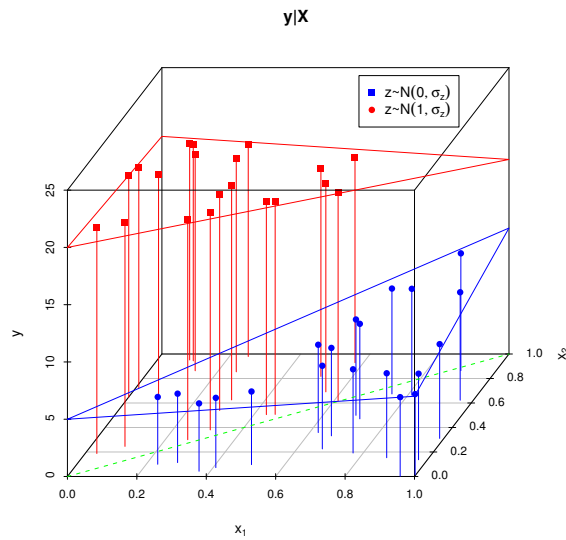


Figure 4.1.1: The plot for (x_1, x_2, y) in Eq. 4.1.1

As designed, the covariate z acts as a random effect, the level of which decides the relationship function between covariate and response. When z has a value more closer to 1 than 0, the covariate-response relationship function is $y = -2x_1 - x_2 + 20$. In the other case, the relationship function is $y = 2x_1 + 4x_2 + 5$. In order to simulate a situation, where we are ignorant of which variable is a random factor and the dataset even does not contain a random factor, we delete z from X . These kind of situations frequently occur in different ways such as inadvertent data collection or need for data transformation. Due to the rapid development of sensor devices, it is quite easy to obtain tons of data. Data are usually collected by someone else than a model-maker without experts' knowledge and may fail to contain a random effect which is crucial in modeling. Even though an expert designs data collection, an important random effect may not be observable directly without data transformation. When

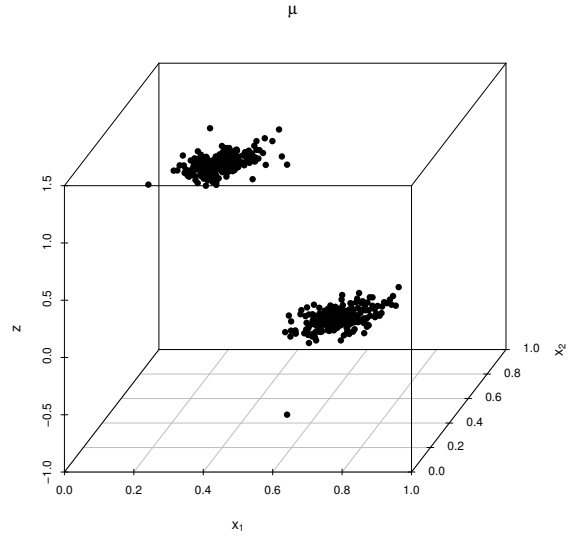


Figure 4.1.2: The plot for the mean parameter (μ) of covariate $X = (x_1, x_2, z)$ over sampling in DPLM

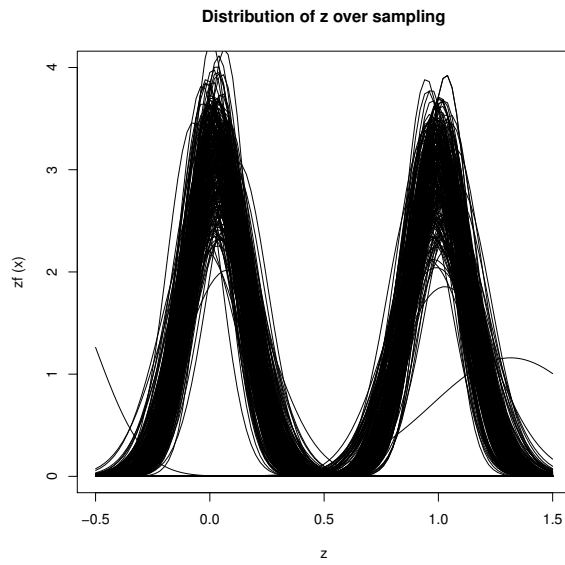


Figure 4.1.3: The distribution of z over sampling in DPLM of covariate $X = (x_1, x_2, z)$

the covariate X consists of only x_1 and x_2 , the sampled points for covariate mean parameter (μ) and their 95% of confidence areas (ellipses) are depicted in (a) of Figure 4.1.4. Like the case of $X = (x_1, x_2, z)$, there are clearly two mixture components and this can be easily verified by investigating the parameter K , denoting the number of components, over sampling (Figure 4.1.5). However, two mixture components share a significant portion of covariate region (overlapped area of two thick ellipses) and more than half of data points are placed in the overlapped area as shown in (b) of Figure 4.1.4. This overlapping causes uncertainty about class identification (a wrong estimate for $F_x(x|\theta_{x,j})$ in the first term of Eq. 2.5.41) and results in bad response estimates. Figure 4.1.6 shows that the response estimates of a model with covariate (x_1, x_2) , denoted by red triangles, deviate much from the reference line (dotted green) while those of a model with covariate (x_1, x_2, z) , denoted by blue squares, are placed on the reference line.

The error, incurred by uncertainty about class identification, is significant when the covariate-response relationships of involved components are significantly different. In this dataset, the maximum value of relative errors (defined differently*) for the model without the random factor z is 52.97% while that for the model with z is 0.32%.

Now we use MDPLM as a tool for modeling. As shown in Figure 4.1.5, the expected number of components is approximately 2 and the size of a validation set is set to 10 by Eq. 3.5.15. Four non-overlapping validation sets are composed by picking elements randomly without replacement. For each validation set, a training data set is obtained by the set difference from the set of all data, denoted by \mathbf{D} . Since $n = |\mathbf{D}| = 40$, we make four training-validation pairs to determine the layer of prediction. After four models are trained and evaluated with their own training-validation dataset pairs upto layer 9, the performance measures $E(cMSE)$ of layers are displayed in Figure 4.1.7. The minimum performance

* $\min(|\hat{y} - y|/y, |\hat{y} - y|/\hat{y})$

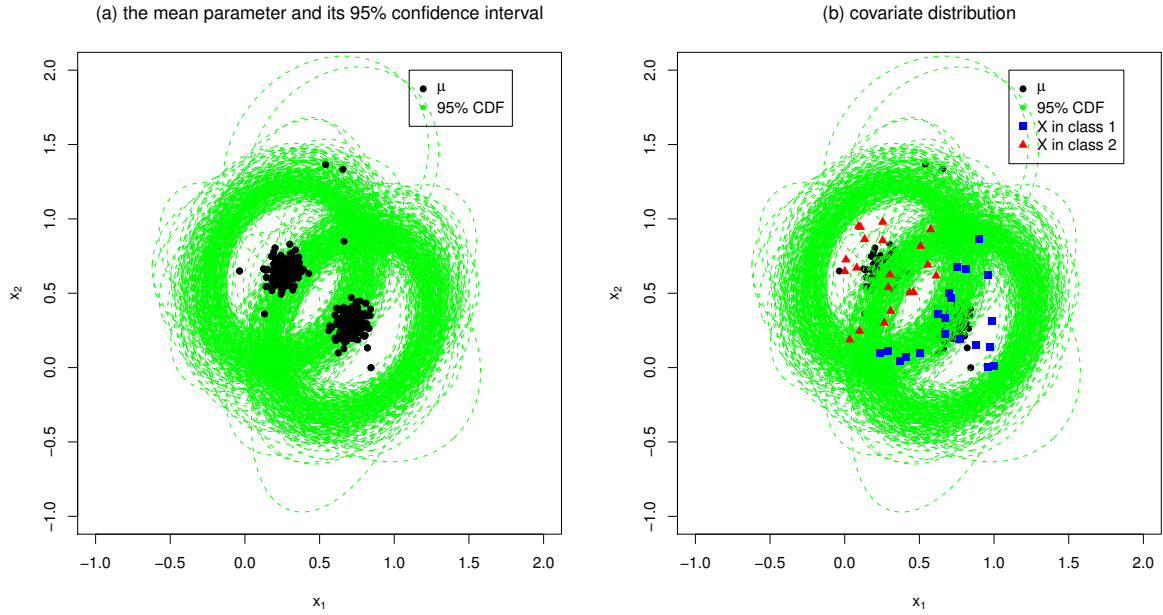


Figure 4.1.4: The overlap of covariate regions: (a) the samples for covariate mean parameter μ of covariate and their 95% confidence areas (ellipses) when $X = (x_1, x_2)$ and (b) the distribution of covariate (x_1, x_2)

measure is achieved at layer 1 and this implies that the response estimates of layer 1 are robust in that the random effect generated at layer 0 is a common characteristic of both training and unseen validation datasets.

The response estimates of two models which were trained with all available data D as a training set are plotted in Figure 4.1.8. The estimates by the final MDPLM, denoted by red crosses, are clearly closer to the green reference line than those of DPLM (equivalent to the single layer MDPLM), denoted by blue circles. Table 4.1 shows numerical comparisons between DPLM and MDPLM about three error measures such as mean square error (MSE*), mean absolute error (MAE[†]), and mean absolute relative error (MARE[‡]).

$$*MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

$$†MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

$$‡MARE = \frac{1}{n} \sum_{i=1}^n \min(|(\hat{y}_i - y_i)/y_i|, |(\hat{y}_i - y_i)/\hat{y}_i|)$$

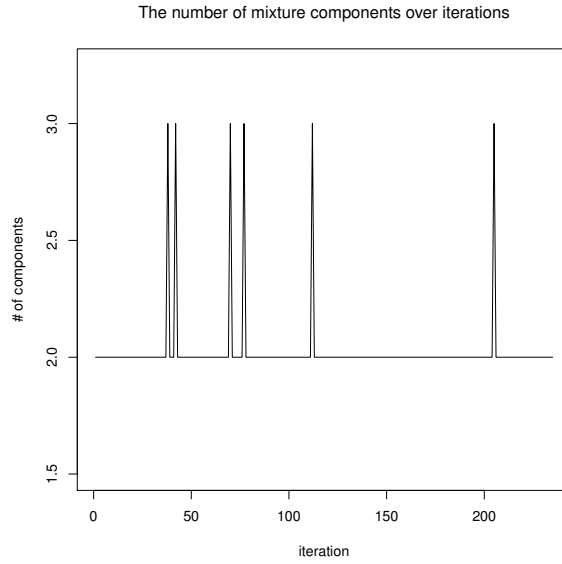


Figure 4.1.5: The number of mixture components (K) over iterations in the model with covariate $X = (x_1, x_2)$

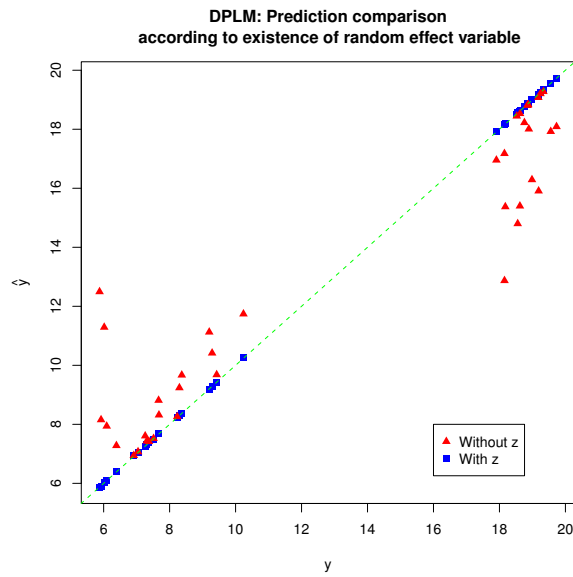


Figure 4.1.6: Prediction comparison of DPLM between $X = (x_1, x_2)$ and $X = (x_1, x_2, z)$

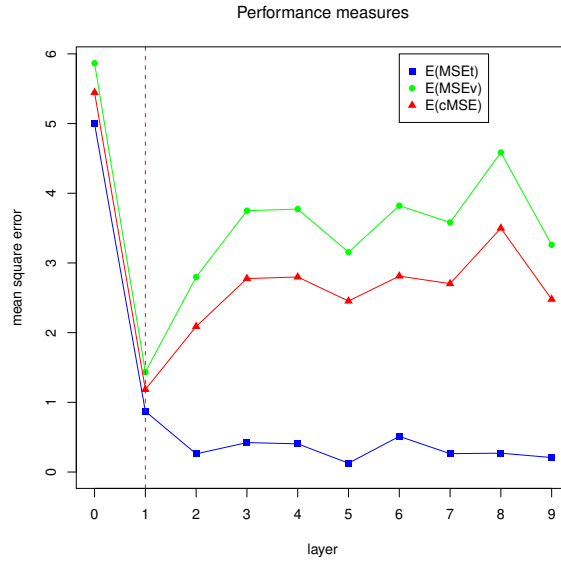


Figure 4.1.7: Performance measures by layers

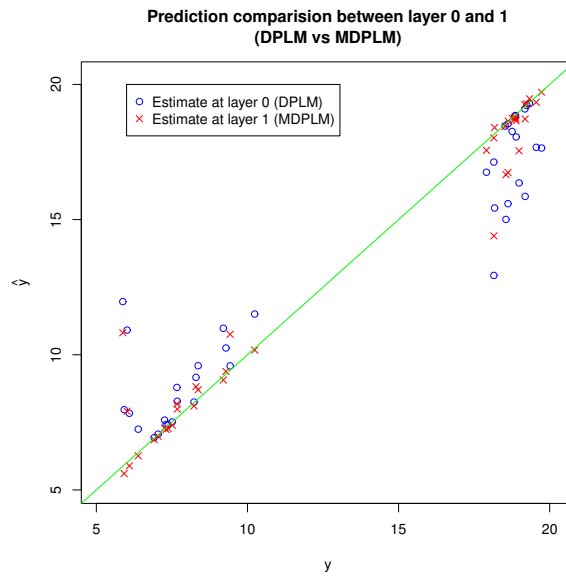


Figure 4.1.8: Comparison of estimates between layer 0 and 1

The dataset (without z) can be regarded as one with a hidden interaction term, which is a nonlinear function of x_1 and x_2 . Since the layer of prediction is layer 1, the training procedure

Table 4.1.1: Comparisons of error measures between DPLM and MDPLM for data of hidden random random effect

	<i>MSE</i>	<i>MAE</i>	<i>MARE</i>
DPLM	4.113	1.312	0.099
MDPLM	1.364	0.565	0.046

of MDPLM is considered as finding a nonlinear function $g(x_1, x_2)$ and regressing responses in a training set with an extended dataset $(x_1, x_2, g(x_1, x_2))$. In prediction time, given the values of x_1 and x_2 , a trained MDPLM estimates both $g(x_1, x_2)$ and response at layer 0 and estimate error with $(x_1, x_2, g(x_1, x_2))$ at layer 1.

4.2 Different Relationships

In general, a regression model assumes a single relationship between covariate and response. Even when a model allows the diversity of relationships by mixture, the forms of relationships are still the same in that they have the same covariate and degree of complexity. On the contrary, MDPLM has flexibility in the diversity of relationships by allowing coexistence of different forms of relationships. Consider a distribution defined in Eq. 4.2.1 and its plot on the (x, y) plane shown in Figure 4.2.1.

$$\begin{aligned} X &= (x, z) \\ x &\sim \mathcal{U}(0, 10) \\ z &\in \{0, 1\} \\ y|X &\sim p_1 I(z = 0) \mathcal{N}(5x, \sigma) + p_2 I(z = 1) \mathcal{N}(x^3, \sigma) \end{aligned} \tag{4.2.1}$$

Different from the previous example, the relationship functions of mixture components have different complexity; f_1 is a linear function, but f_2 is a cubic as shown in Eq. 4.2.2. Because of linearity assumption of response function in DPLM, the cubic curve is fitted with piecewise linear functions, that is, there exist more than one component which composes the cubic function. Figure 4.2.2 shows the sampled means and the 95% confidence areas of components' covariate functions, denoted by black dots and green ellipses (almost flat on x -axis) respectively, at iteration 3000. Since z is discrete, the variance of z in DPLM (the first layer of MDPLM) is very narrow comparing with that of x . The cubic function consists of the components the means of which are placed around $z = 1$. In contrast to the rest of components around $z = 0$, the means around $z = 1$ are distributed widely, which implies that the linear covariate-response assumption does not hold well and more components

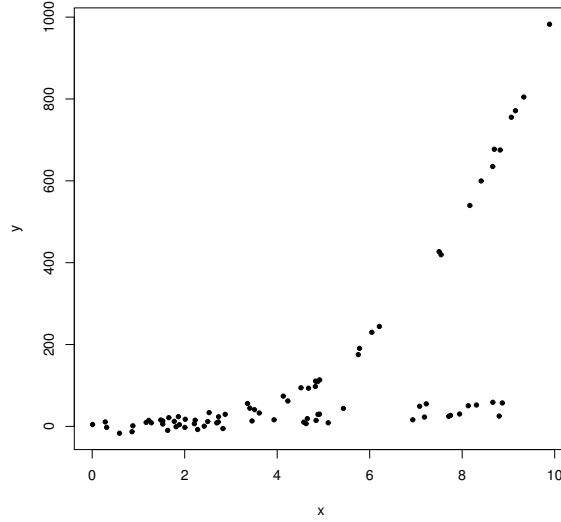


Figure 4.2.1: The plot for (x, y) of Eq. 4.2.1

are needed to fit. Since the observation points of f_1 are not clearly separable from those of f_2 in the interval $(0, 5)$ of x (see Figure 4.2.1), a component around $z = 0.5$ occurs, but its effect is insignificant to the response estimate because the covariate function value of other components are much greater.

$$f_1 : y = 5x + \sigma \quad \text{if } z = 0 \tag{4.2.2}$$

$$f_2 : y = x^3 + \sigma \quad \text{if } z = 1.$$

MDPLM has the minimum $E(cMSE)$ at layer 1 whereafter the validation errors increase rapidly (shown in Figure 4.2.3). For the cubic function, since the rate of response over x (that is, $\frac{\partial y}{\partial x}$) is still a square function of x , the response increases rapidly as x increases. Even with piecewise linear modeling, this asymmetric complexity cannot be dealt with well by

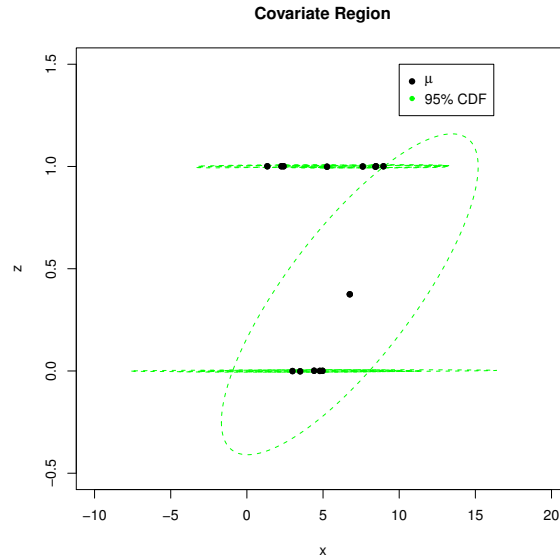


Figure 4.2.2: The covariate region of DPLM for Eq. 4.2.1 at iteration 3000

DPLM. Such inability of DPLM is shown in Figure 4.2.4 where the response estimates of f_2 (denoted by blue circles) deviate much from the green reference line in the interval (5, 10) of x . MDPLM overcomes this problem by increasing layers and fits itself to data complexity. The error reduction of MDPLM over DPLM is summarized in Table 4.2.1.

Table 4.2.1: Comparisons of error measures between DPLM and MDPLM for data of different relationships

	<i>MSE</i>	<i>MAE</i>	<i>MARE</i>
DPLM	2948.54	29.27	0.44
MDPLM	615.73	16.00	0.45

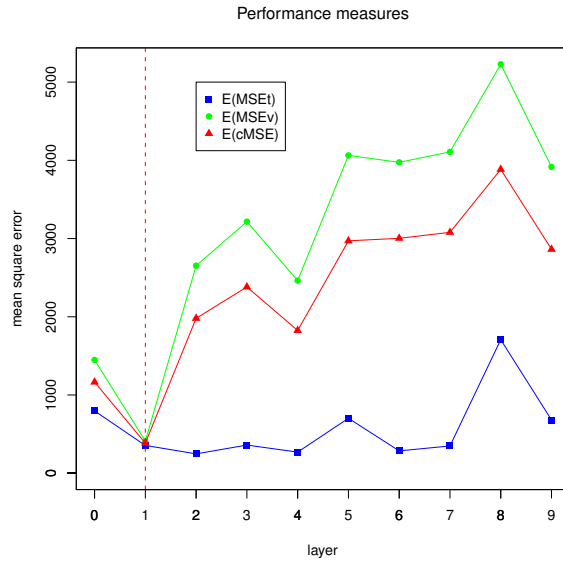


Figure 4.2.3: Performance measures by layers

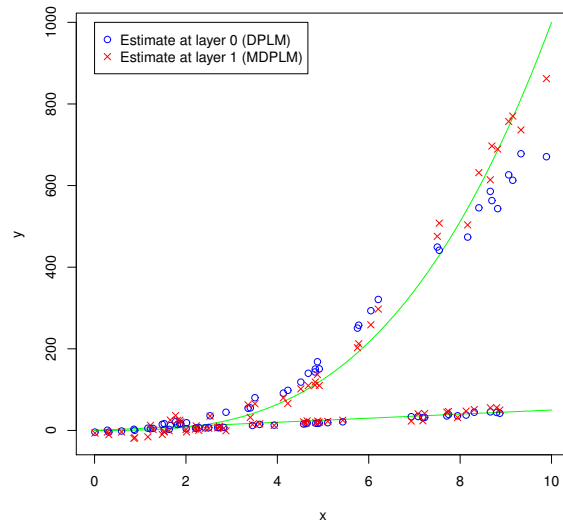


Figure 4.2.4: Comparison of response estimate between DPLM and MDPLM

4.3 Heteroscedasticity

In a regression, the covariate-response relationship is said to be homoscedastic when the variance of error term is constant. On the other hand, when the variance of error term changes or depends on the covariate, the relationship is said to be heteroscedastic. Heteroscedasticity is often a by-product of violation of assumptions such as model misspecification. As a common cause of heteroscedasticity, subpopulation differences end up with biased error measures. Like DPLM, MDPLM can handle such a heteroscedasticity problem from subpopulation differences by finding a random effect when one of covariates is related with the random effect. Moreover, MDPLM detects a hidden random effect of more complicated forms (e.g., nonlinear or interacted) described in Section 4.1 and results in less biased error measurements.

In heterogeneous population data, it is not reasonable to assume that error is an independent and identically distributed (*i.i.d.*) random variable throughout classes. A simple relaxation of the assumption is that error is *i.i.d.* within class. For instance, let e_{ij} be the error of observation j in class i and σ_i be the standard deviation of class j . The normality assumption of *i.i.d.* error within class can be represented by

$$e_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_i^2), \quad (4.3.1)$$

which means that all elements in a class share the common variance of error.

As another way of relaxation, error is assumed to be independent not (necessarily) identically distributed (*i.n.i.d.*). This kind of error distributions is easily observed in growth data where the variance of error changes over covariate. Such heteroscedasticity can be detected by statistical tests such as Breusch-Pagan test [34] or White's test [35], but the form of error and involved covariate should be specified.

Consider data points of the distribution described in Eq. 4.3.2; the distribution has an *i.n.i.d.* random error. As illustrated in Figure 4.3.1, the variance of error e is the function of covariate x and increases along with x . Assume that we are ignorant of both the covariate-response relationship and the form of error distribution. Even though the relationship function were assumed to be known, the form of error distribution could be still vague in heterogeneous population data because Figure 4.3.1 can be thought to be generated by two or more similar, but different covariate-response relationships with a common *i.i.d.* error. However, DPLM or MDPLM can estimate responses via sampling parameters, not estimating, and it is more reasonable to consider all possible cases than to choose the one most likely to occur. Moreover, since an estimated response is a weighted average of response estimates by parameter significance, estimation tends to be robust.

$$y = \log[1 + x + e(x)] \quad (4.3.2)$$

$$e(x) \sim \mathcal{N}(0, x/8)$$

Modeling with MDPLM shows that the performance measure $E(cMSE)$ drops significantly at layer 1 and has a minimum value at layer 2 (Figure 4.3.2). The relatively significant reduction of $E(cMSE)$ at layer 1 indicates that the assumptions of DPLM are violated greatly. During the parameter sampling at the first layer (layer 0), the expected number of mixture components is about 3.4 (see Figure 4.3.3) and there are three dominant components which are centered around at $x=4$, 20, and 42. Let us call the components C_1 , C_2 and C_3 from left to right, and the corresponding covariate regions X_1 , X_2 and X_3 , respectively. According to within-class *i.i.d.* assumption on random errors in DPLM, it can be viewed that the quadratic function $(\frac{x}{8})^2$ of error variance is decomposed into weighted three constant error variances

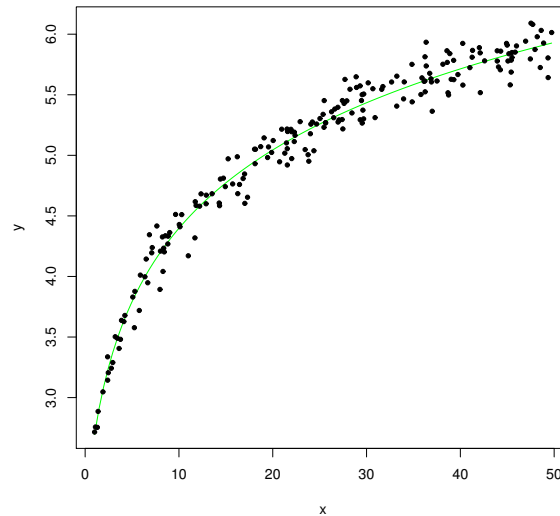


Figure 4.3.1: The plot for $y = \log[1 + x + e(x)]$: the variance of error increases as x increases.

and their weights are the functions of covariate x .

The comparison of results between DPLM and MDPLM is presented graphically in Figure 4.3.4 and numerically in Table 4.3.1. The deviation of response estimate by DPLM from the reference line (green dotted in Figure 4.3.4) increases as x increases. This can be easily explained when we see Figure 4.3.6; two groups of regression lines are clearly identified as two thick overlapped lines and the reference line (dotted green) is placed in the lower part of such thick lines. The displacement of reference line from the center of thick line in range $x=(40,50)$ indicates that response estimates by DPLM would be overestimated because only the component C_3 is dominant in that range (shown in Figure 4.3.5). On the contrary, the response estimates of MDPLM are very close to the reference line, which shows that MDPLM handles the nonlinear covariate-response relationship and heteroscedasticity of the data simultaneously.

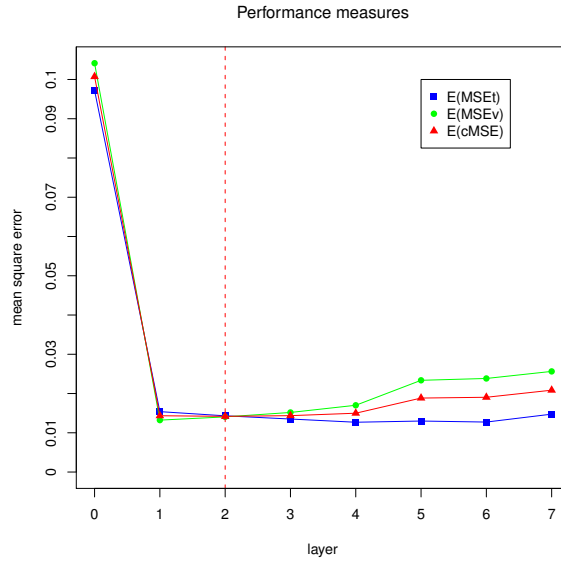


Figure 4.3.2: The performance measures of MDPLM over layers: a minimum $E(cMSE)$ is achieved at layer 2.

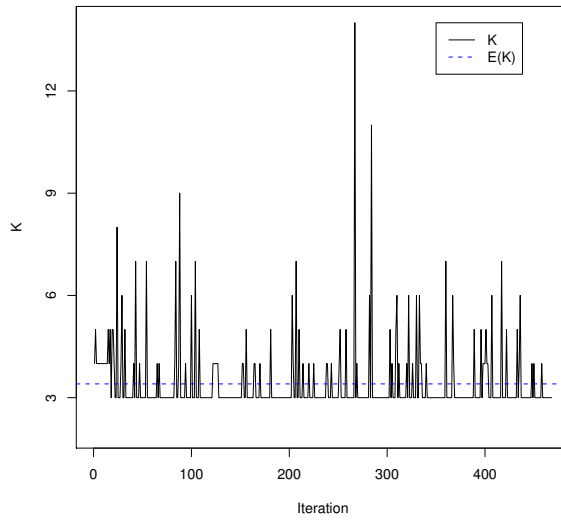


Figure 4.3.3: The plot for the number of components (K) over parameter sampling: $E(K) \approx 3.4$

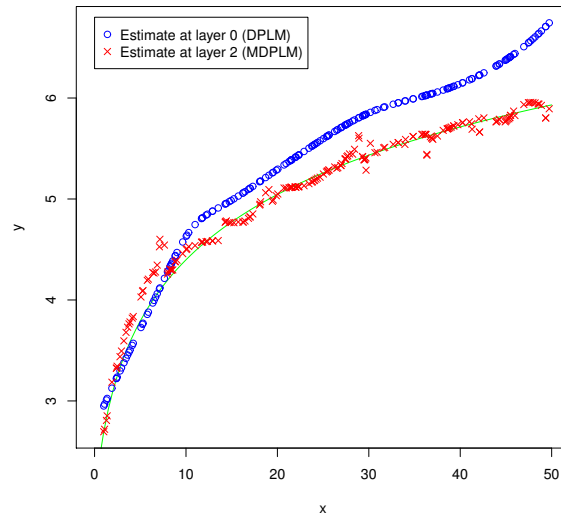


Figure 4.3.4: Comparison of response estimates between DPLM and MDPLM.

Table 4.3.1: Comparisons of error measures between DPLM and MDPLM for data of heteroscedasticity

	<i>MSE</i>	<i>MAE</i>	<i>MARE</i>
DPLM	0.151	0.336	0.060
MDPLM	0.020	0.108	0.022

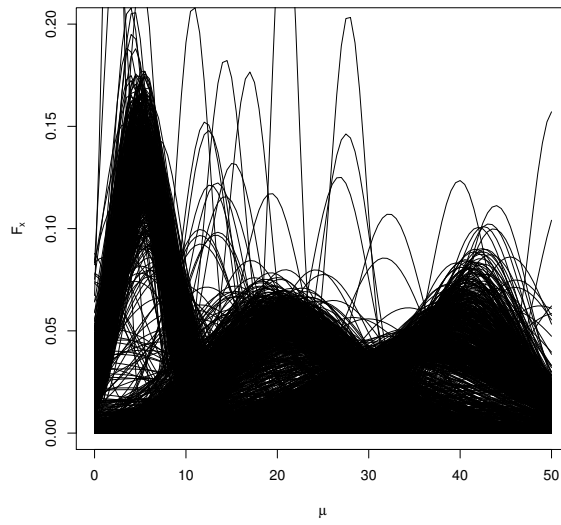


Figure 4.3.5: The plot for $\mathcal{N}(\mu, s^{-2})$ over sampling

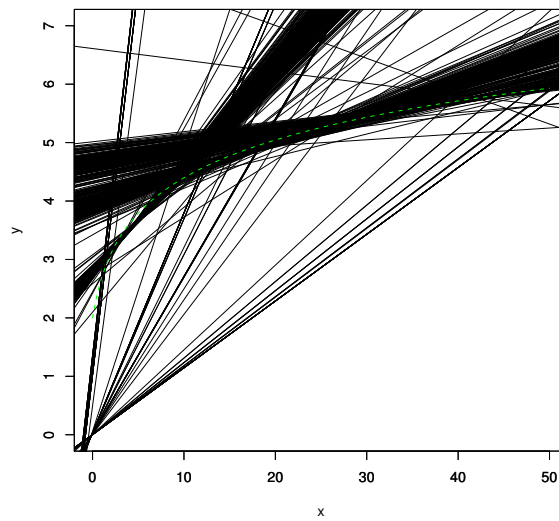


Figure 4.3.6: Sampled regression lines

5.1 MDPLM for Preliminary Engineering Cost

In this section, we will describe how to use MDPLM for data of unknown distribution.

5.1.1 Background

The *preliminary engineering* (PE) is a process in the phases of project and has received gaining attention over decades in the field of construction. Its importance cannot be underestimated because it is not only closely related to the plan of resource allocation, but it takes a significant portion of total construction cost. Therefore, the expense on PE, called PE cost, also must be an influential factor on the success of final project by balancing between quality and efforts of project design. However, estimating PE cost is challenging because its characteristics are

not fully understood.

In this section, we construct a prediction model to estimate PE ratio, the ratio of PE cost over total construction cost, with data of highway bridge projects, described in the next section. The main difficulty of modeling comes from uncertainty about characteristics of data: how covariates relate to response, what distribution error terms have, how many different covariate-response relationships exist, which covariates have random effects, whether hidden random effects exist, how a covariate interacts with other covariates, etc. In modeling, we assume that there is no known information about the dataset except that it may be heterogeneous. Since DPLM can deal with heterogeneity in some degree, we compare the results from an MDPLM and a DPLM, trained by the same dataset, in terms of three error measurements such as mean square error (*MSE*), mean absolute error (*MAE*) and mean absolute relative error (*MARE*).

5.1.2 Data Description

The dataset consists of 505 bridge projects let for construction by *North Carolina Department of Transportation* (NCDOT) between January 1999 and June 2008. The ten data sources, listed in Table 5.1.1, were used to populate the project database.

For constructing a prediction model, 13 independent variables were selected as covariates with the response, the percentage scaled PE ratio ($= \text{PE_RATIO} \times 100$), as shown in Table 5.1.2.

Table 5.1.1: Data sources for bridge projects

1. NCDOT Online Bid Tabulations & Annual Bid Averages Summary
2. NCDOT Pre-2002 Project Management Data System (obsolete mainframe system)
3. NCDOT Post-2002 Project Management Data System (SAP based)
4. NCDOT 12-Month Projected Letting List
5. NCDOT National Bridge Inventory System Data (NBIS)
6. NCDOT State Transportation Improvement Plan (STIP)
7. NCDOT Transport Program Modification - Project Type Coding
8. NCDOT Online Construction Plans
9. NCDOT Board of Transportation Minutes and Funding Authorizations
10. North Carolina State Publications Clearinghouse

Table 5.1.2: The list of variables for bridge model: $x_{1:13}$ are the covariates and y is the response

Variable	Names	Description
x_1	S147_ST_B_C	Structure designation (bridge=1 or culvert=2)
x_2	RW	Proportion of total construction cost that can be attributed to the roadway portion (numerical)
x_3	ST	Proportion of total construction cost that can be attributed to the structure components (numerical)
x_4	S102_ROAD_SYSTEM	Designation for arterial, collector, or local route system carried by structure (arterial = 1, collector=2, local route system = 3)
x_5	NEPA_DOC	Complexity of the environmental documentation required for each project from least to most (Categorical exclusion = 1 or Programmatic categorical exclusion = 2)
x_6	LEN	Total length of the project in miles (numerical)
x_7	PLAN_RESP	Party responsible for ensuring delivery of the NEPA document (Department of transportation = 1 or Private engineering firm =2)
x_8	GEO_AREA	Project location (Coast=1, Mountain=2, Piedmont=3, V mountain=4)
x_9	N26_R_U	Classification of route (Rural=1 or Urban=2)
x_{10}	TIP_COST	Updated construction cost estimate (numerical)
x_{11}	ROW_COST	Costs attributed to right of way (numerical)
x_{12}	ROW_RATIO	Ratio of right of way costs over estimated construction costs (numerical)
x_{13}	N45_NUM_MAIN_SPAN	Number of spans in main structure (numerical)
y	PE_RATIO %	Percentage of preliminary engineering costs over estimated construction costs

5.1.3 Procedure and Results

From the DPLM, trained with 505 bridge project observations, the expected number of mixture components was approximately 10 and, thus, the size of a validation set was set to 25 by Eq. 3.5.15. For a less bias performance measure, 10 training-validation dataset pairs were collected in the way described in Section 3.5.4. The designed noise rates of layers were determined by searching 10 points, equally separated in the range $(0, 0.1]$, sequentially from the first layer. We continued searching to layer 6 in order to verify error transition and the sequence of noise rates found is listed in Table 5.1.3.

Table 5.1.3: The designed noise rates of bridge projection

layer (ℓ)	0	1	2	3	4	5	6
designed noise rate (η_ℓ)	0.1	0.02	0.06	0.06	0.04	0.08	0.08

Figure 5.1.1 plots the averaged error transitions of three basic error measurements ($E(MSE)$, $E(MAE)$ and $E(MARE)$) over 10 training-validation dataset pairs for both training and validation data. As shown in Figure 5.1.1 (a), the layer 4 turned out be robust for prediction by achieving a minimum $E(cMSE)$ and, thus, the layer of prediction was set to 4. Since $E(MSE)$ and $E(MAE)$ for validation data leap up at the layer 5, the layer 4 is a reasonable layer for robust prediction. The benefit of using MDPLM over DPLM (equivalent to a single layer MDPLM) can be clearly observable numerically in Table 5.1.4. The training errors have reduced by a factor of 4.533, 2.670, 2.873 for mean square error, mean absolute error, and mean relative error, respectively. The corresponding validation errors have also reduced by a factor of 3.356, 10.839 and 1.523, respectively.

Once the layer of prediction was determined, the final model was trained with all data available (505 bridge construction projects) along with the designed noise rates in Table 5.1.3

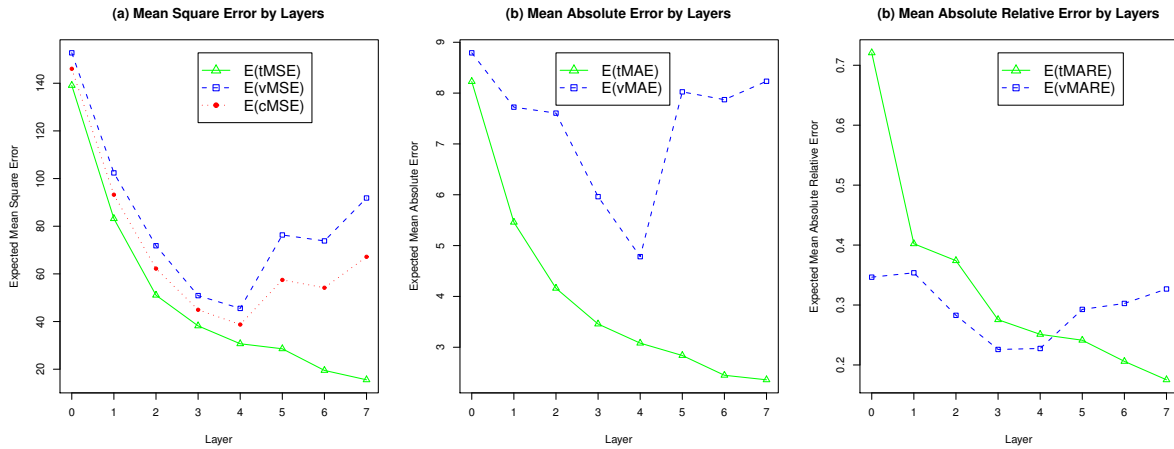


Figure 5.1.1: Determining the layer of prediction for the bridge construction model: (a) expected mean square error, (b) expected absolute error, and (c) expected absolute relative error over 10 training-validation dataset pairs; MSE, MAE and MARE denote mean square error, mean absolute error, and mean absolute relative error in the same order; subscripts ‘t’ or ‘v’ indicates a training or validation data over which error measure is computed, respectively; the plot (a) shows that $E(cMSE)$ has the minimum at layer 4

Table 5.1.4: Comparison of error measurements for the bridge project model between layer 0 and 4

base measure of expected value	layer 0		layer 4	
	training	validation	training	validation
mean square error	139.090	152.757	30.679	42.521
mean absolute error	8.227	8.792	3.081	4.782
mean relative error	0.721	0.346	0.251	0.224

upto layer 4. The response estimates of layers in Figure 5.1.2 illustrate that the model of layer 4 fits well the bridge construction dataset, and the robustness of its response estimation for unseen data is guaranteed by cross-validation.

Table 5.1.5: The error measurements of the final bridge MDPLM for preliminary engineering cost ratio estimation

mean square error	mean absolute error	mean relative absolute error
31.624	3.013	0.208

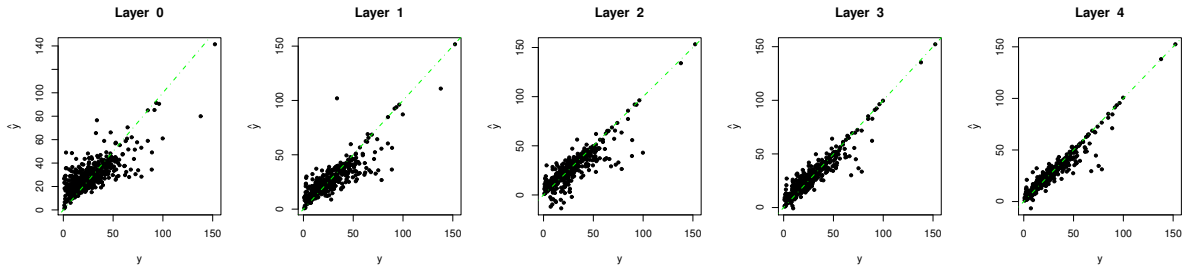


Figure 5.1.2: The layer-wise predictive results of the bridge MDPLM trained with 505 construct projects: y is the response in the data set and \hat{y} is its estimate

5.2 MDPLM for Capsular Penetration in Prostate Cancer

In this section, we will test the usefulness of MDPLM as a classifier. The data used for modeling are known to be well-fitted by a logit model and frequently used for pedagogical purpose.

5.2.1 Background

Prostate cancer, which starts in the prostate gland, is the most common cause of death in men over age 75 [36]. As for a screening test, the *prostatic-specific antigen* (PSA) blood test is often done to measure the serum prostate-specific antigen level. The *digital rectal exam* (DRE) is one of the diagnostic tests used to exam abnormality of the prostate gland physically. The diagnosis can be confirmed only by prostate biopsy, where tissue samples from the prostate are viewed underneath a microscope. The results of prostate biopsy report the Gleason grade which indicates the degree of cancerous cells by examining how different the patterns, sizes, and shapes of sampled cells are from healthy prostate cells. The Gleason score represents the aggressiveness of prostate cancer, obtained by adding Gleason grades of the two most common patterns of prostate cancer cells. Therefore, the higher Gleason score

means that the cancer is more likely to spread beyond the prostate gland.

The prostate capsule is the membrane which surrounds the prostate gland. Categorized as no capsular involvement, capsular invasion or capsular penetration, the tumoral involvement to the capsule represents the progress of the cancer and becomes an important factor for prognosis [37]. In this section, we construct an MDPLM to predict the capsular penetration in prostate cancer. In modeling, we do not assume anything about data distribution, but we modify the discrete values of the response to prevent the mean square errors of prediction from being underestimated.

5.2.2 Data Description

The dataset [38] contains the diagnosis record of 376 patients with prostate cancer. The data are a subset of variables from a study of prostate cancer at *The Ohio State University Comprehensive Cancer Center*. The original dataset had the record of 380 patients, but we removed 4 of them from the record because of missing data. Also, the binary (i.e., 0 or 1) response variable representing the capsular penetration was changed to have 10 or 20 because the performance measure of MDPLM is a weighted sum of mean square errors. The description of variables used is presented in Table 5.2.1. Of the 376 patients, 151 had the prostate cancer of capsular penetration and 225 had the cancer of either the absence of capsular involvement or the capsular invasion alone without penetration.

Table 5.2.1: The list of variables for the prostate capsule penetration model: $x_{1:7}$ are the covariates and y is the response

Variable	Names	Description
x_1	AGE	Age of patient (numerical)
x_2	RACE	Race of patient (White=1 or Black=2)
x_3	DPROS	Result of the digital rectal exam (No nodule=1, Unilobar nodule (left)=2, Unilobar nodule (right)=3, or Bilobar nodule =4)
x_4	DCAPS	Detection of capsular involvement in rectal exam (No=1 or Yes=2)
x_5	PSA	Prostatic specific antigen value (numerical mg/ml)
x_6	VOL	Tumor volume obtained from ultrasound (numerical cm ³)
x_7	GLEASON	Total Gleason score (numerical, integer in [0,10])
y	CAPSULE	Tumor penetration of prostatic capsule (No penetration=10 or Penetration=20 for DPLM and MDPLM) (No penetration=0 or Penetration=1 for logit models)

5.2.3 Procedure and Results

Different from the previous application, the response CAPSULE is a binary variable and the prediction of model is classification. Due to the known number of classes, we reserve 113 records for testing purpose (30% data from each class). Therefore, we use the rest 263 (70%) data to make three models using DPLM, MDPLM, and logit regression. The performance of models will be compared for 113 testing data.

The evaluation of classification performance is based on the counts of test records correctly or incorrectly predicted by a model. A confusion matrix displays the number of records, denoted by n_{ij} , from class i to be predicted to class j . Hence, the accuracy (or error rate) of a model can be computed from a confusion matrix by using Eq. 5.2.1 (or Eq. 5.2.2). Most classification techniques or algorithms seek a model which attain the highest accuracy, or equivalently the lowest error rates, and we also follow the criterion for the best model.

Table 5.2.2: Confusion matrix

		Predicted class	
		class=1	class=0
Actual class	class=1	true positive (n_{11})	false negative (n_{10})
	class=0	false negative (n_{01})	true negative (n_{00})

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} & (5.2.1) \\
 &= \frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}}
 \end{aligned}$$

$$\begin{aligned}
\text{Error rate} &= \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} \\
&= \frac{n_{10} + n_{01}}{n_{11} + n_{10} + n_{01} + n_{00}} \\
&= 1 - \text{Accuracy}
\end{aligned} \tag{5.2.2}$$

MDPLM

From the DPLM, trained with 263 patient observations, the expected number of mixture components was 4, so the size of a validation set was set to 33 by Eq. 3.5.15. We collected 4 training-validation dataset pairs in the way of Section 3.5.4. The designed noise rates of a layer were determined by searching 10 points, equally separated in the range $(0, 0.1]$, sequentially from the first layer. We continued searching upto layer 4 to see the error transition and the sequence of noise rates found is listed in Table 5.2.3.

Table 5.2.3: The designed noise rates of the prostatic capsular penetration MDPLM

layer (ℓ)	0	1	2	3	4
designed noise rate (η_ℓ)	0.04	0.02	0.02	0.06	0.1

Figure 5.2.1 shows the averaged error transitions over 4 training-validation dataset pairs. Our performance measure $E(cMSE)$ has the minimum value at the layer 3, but we set the layer 1 as the layer of prediction according to the way described in Section 3.5.4.

We trained MDPLM with 263 patient observations upto the layer 5 for comparison purpose and tested with 113 testing data. Since the predicted outputs from MDPLMs are continuous, we convert them to binary response values by changing values less 15 to 0 and values greater than 15 to 1. For 113 testing data, the resulting confusion matrices and their error rates along with different layers of prediction are presented in Table 5.2.4. The model

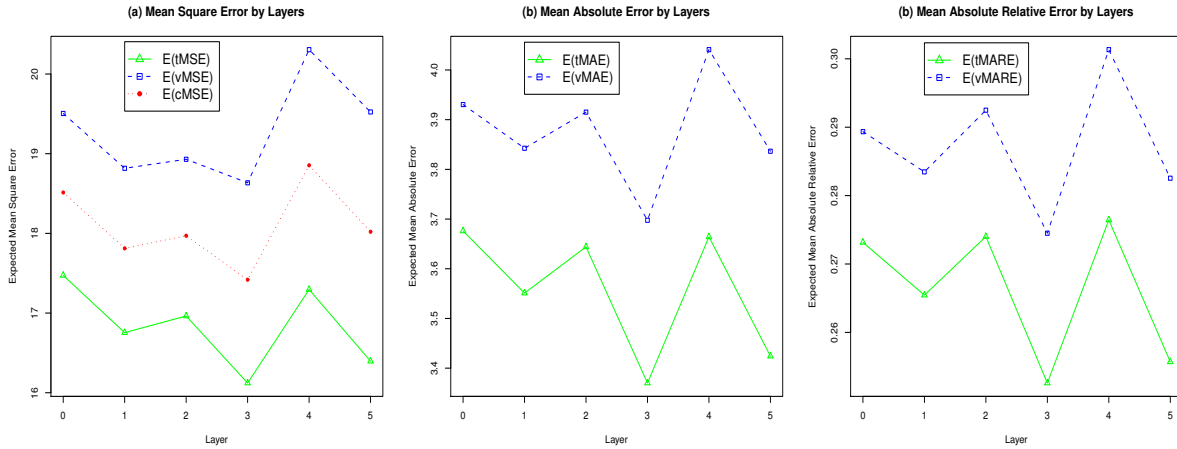


Figure 5.2.1: Determining the layer of prediction for the prostate cancer model: (a) expected mean square error, (b) expected absolute error, and (c) expected absolute relative error over 10 training-validation data set pairs; MSE, MAE and MARE denote mean square error, mean absolute error, and mean absolute relative error in the same order; subscripts ‘t’ or ‘v’ indicates a training or validation data over which error measure is computed, respectively; the layer of prediction is set to 1 even though the layer 3 has a minimum $E(cMSE)$.

that has the layer 1 for prediction is one of three models which have the minimum error rate. Due to the preference to the simpler model, that model is selected and the result corresponds to the selection of model in the MDPLM procedure.

Table 5.2.4: Confusion matrices of MDPLMs with the different layers of prediction; pLayer denotes the layer of prediction; CP and NCP denote the presence and the absence of the capsular penetration, respectively.

		Predicted class											
		pLayer = 0		pLayer = 1		pLayer = 2		pLayer = 3		pLayer = 4		pLayer = 5	
		CP	NCP	CP	NCP	CP	NCP	CP	NCP	CP	NCP	CP	NCP
Actual class	CP	33	12	36	9	42	3	35	10	35	10	33	12
	NCP	11	57	12	56	29	39	11	57	11	57	11	57
Accuracy		0.796		0.814		0.717		0.814		0.814		0.796	
Error rate		0.204		0.186		0.283		0.186		0.186		0.204	

logit model

Since the response is a binary variable, a natural choice of modeling is the *logit model* (or *logistic model*). We make a logit model with 263 patient observation to classify the prostatic capsular penetration and test with 113 testing data.

As shown in Eq. 5.2.3 and Eq. 5.2.4, given a binary response variable y_i , the *logit model* defines the probability of success, denoted by π_i , (i.e., when $y_i = 1$) as the nonlinear function of the linear predictor z_i .

$$\begin{aligned}\pi_i &= p(y_i = 1|x_i) \\ &= \frac{e^{z_i}}{1 + e^{z_i}}\end{aligned}\quad (5.2.3)$$

$$\begin{aligned}z_i &= \text{logit}(\pi_i) \\ &= \log\left(\frac{\pi_i}{1 - \pi_i}\right) \\ &= \beta_0 + \beta_1 x_{i1} + \dots + \beta_d x_{id}\end{aligned}\quad (5.2.4)$$

In addition, the logit model assumes that (1) the response y_i is independently distributed, (2) the distribution of y_i is binomial distribution (i.e., $y_i \sim \text{Bin}(n_i, \pi_i)$), and (3) the logit of the response probability has a linear relationship with covariate as shown in Eq. 5.2.4.

We made 7 logit models by deleting covariates according to their statistical significances (e.g., *p-value*) and tested with 113 testing data. The results of their performances are listed in Table 5.2.5. The logit models do not show much difference in performance in that their training performance measures (*AICs*) and testing performance measures (error rates) are similar to those of one another. Considering model simplicity, logit 7 will be selected as the best model among them.

Table 5.2.5: Logit models for prostatic capsular penetration

Model	Covariates	AIC	Testing error rate
logit1	AGE, RACE, DPROS, DCAPS, PSA, VOL, GLEASON	290.81	0.204
logit2	RACE, DPROS, DCAPS, PSA, VOL, GLEASON	289.88	0.204
logit3	AGE, DPROS, DCAPS, PSA, VOL, GLEASON	288.83	0.195
logit4	AGE, RACE, DPROS, PSA, VOL, GLEASON	288.81	0.195
logit5	AGE, RACE, DPROS, DCAPS, PSA, GLEASON	291.62	0.186
logit6	RACE, DPROS, DCAPS, PSA, GLEASON	291.12	0.195
logit7	AGE, RACE, DPROS, PSA, GLEASON	289.65	0.186

Results

We tested the usefulness of MDPLM as a classifier with the data of known distribution. Comparing the MDPLM, which has the layer 1 as the layer of prediction, with the logit models, the MDPLM has the same error rate as the logit models of the lowest error rate. Even without any assumption about the data, MDPLM produced a robust model in prediction.

Summary and Conclusions

In this chapter, we summarize and conclude the dissertation. In Section 6.1 the MDPLM procedure is summarized in a step-by-step manner, and in Section 6.2 MDPLM is discussed in two points of view such as model assumption and bias-variance tradeoff. In Section 6.3 the contributions of MDPLM are discussed in detail. Finally, we discuss possible future studies with elaboration in Section 6.4.

6.1 Summary of MDPLM Procedure

In this section, the MDPLM procedure is summarized in a step-by-step fashion for a clear view to how the concepts in Chapter 3 are integrated. We introduce more notations, which are not avoidable in integration, and change the meanings of some notations previous used. Also, for the sake of simplicity, we regard a model, such as DPLM, oDPLM and MDPLM,

as a procedural block with inputs in steps of MDPLM procedure. For instance, $DPLM_{\ell,m}$, denoting a model trained with a dataset \mathbf{D}_{t_m} , is regarded as a functional block with layer ℓ and index m as inputs. The MDPLM procedure will follow right after notations and it consists with 12 steps where steps 8 to 11 are responsible for layer generation and robustness check. The procedure is based on the sequential search for decisions on designed noise rates and the layer of prediction in Section 3.5.5.

Notations for MDPLM procedure

\mathbf{D}	a given dataset.
N	the number of observations in \mathbf{D} .
N_v	the size of a validation set.
N_t	the size of a training set.
K	the parameter of DPLM representing the degree of heterogeneity.
$K^{(t)}$	a realization of K sampled from its posterior distribution at iteration t .
M	the number of training-validation dataset pairs.
\mathbf{D}_{v_m}	the m_{th} validation dataset.
\mathbf{D}_{t_m}	the m_{th} training dataset.
\mathbf{D}_m	the m_{th} training-validation dataset pair $(\mathbf{D}_{t_m}, \mathbf{D}_{v_m})$.
η_ℓ	a designed noise rate introduced to $DPLM_{\ell,m}$ to construct $oDPLM_{\ell,m}^{\eta_\ell}$. (shared by all training-validation dataset pairs)
$\eta_{0:(\ell-1)}^*$	the sequence of designed noise rates found upto layer ℓ to minimize $E(cMSE_\ell)$. (shared by all training-validation dataset pairs)
$MDPLM_{\ell,m}$	MDPLM of $(\ell + 1)$ layers with $\eta_{0:(\ell-1)}^*$ for \mathbf{D}_m .
$MDPLM_{\ell,m}^{\eta_{\ell-1}}$	MDPLM of $(\ell + 1)$ layers with a designed noise rate sequence $(\eta_{0:(\ell-2)}^*, \eta_{\ell-1})$ for \mathbf{D}_m .

$DPLM_{\ell,m}$	DPLM at layer ℓ in $MDPLM_{\ell,m}$.
$oDPLM_{\ell,m}$	$oDPLM$ at layer ℓ in $MDPLM_{(\ell+1),m}$. ($oDPLM$ generated from $DPLM_{\ell,m}$ by a designed noise rate η_ℓ^*)
$oDPLM_{\ell,m}^{\eta_\ell}$	$oDPLM$ generated from $DPLM_{\ell,m}$ by a designed noise rate η_ℓ .
$DPLM_{(\ell+1),m}^{\eta_\ell}$	DPLM constructed from $DPLM_{\ell,m}$ and $oDPLM_{\ell,m}^{\eta_\ell}$
$cMSE_{\ell,m}$	the performance measure of $MDPLM_{\ell,m}$.
$cMSE_{\ell,m}^{\eta_\ell}$	the performance measure of $MDPLM_{\ell,m}^{\eta_\ell}$.
$E(cMSE_\ell^{\eta_\ell})$	the overall performance measure for layer ℓ with a designed noise rate sequence $(\eta_{0:(\ell-2)}^*, \eta_{\ell-1})$ over all the training-validation dataset pairs.
$E(cMSE_\ell)$	the overall performance measure for layer $(\ell + 1)$ with the designed noise rate sequence $\eta_{0:(\ell-1)}^*$ over all the training-validation dataset pairs.

The MDPLM procedure

1. Construct a DPLM with \mathbf{D} .
2. Compute the expected value of the degree of heterogeneity.

$$E(K) = \frac{1}{T} \sum_{t=1}^T K^{(t)}$$

3. Determine the size of a validation set.

$$N_v = \left\lceil \frac{N}{2E(K)} \right\rceil$$

4. Make M training-validation dataset pairs from \mathbf{D} .

(M is arbitrarily chosen so that $1 \leq M \leq \lfloor N/N_v \rfloor$, but it is recommended to choose a number bigger than 4)

(a) Sample M validation datasets such that $|\mathbf{D}_{v_m}| = N_v, \forall m$, and $\bigcap_{m=1}^M \mathbf{D}_{v_m} = \emptyset$.

(b) Set $\mathbf{D}_{t_m} = \mathbf{D} \setminus \mathbf{D}_{v_m}, \forall m$.

5. Construct $\text{DPLM}_{0,m}$ with $\mathbf{D}_{t_m}, \forall m$. ($\text{MDPLM}_{0,m} \equiv \text{DPLM}_{0,m}$)

6. Compute $E(cMSE_0)$ with all training-validation dataset pairs.

$$E(cMSE_0) = \frac{1}{M} \sum_{m=1}^M cMSE_{0,m}$$

7. Set $\ell = 1$.

8. Execute the following for $m = 1, \dots, M$ and for $\eta_{\ell-1}$ on the grid of designed noise rate.

(a) Construct $\text{oDPLM}_{\ell-1,m}^{\eta_{\ell-1}}$.

(b) Construct $\text{DPLM}_{\ell,m}^{\eta_{\ell-1}}$ for each pair of $\text{DPLM}_{\ell-1,m}$ and $\text{oDPLM}_{\ell-1,m}^{\eta_{\ell-1}}$.

(c) Construct $\text{MDPLM}_{\ell,m}^{\eta_{\ell-1}}$ from $\text{MDPLM}_{(\ell-1),m}$ and $\text{DPLM}_{\ell,m}^{\eta_{\ell-1}}$.

9. Compute $E(cMSE_{\ell}^{\eta_{\ell-1}})$ for each $\eta_{\ell-1}$ on the grid of designed noise rate.

$$E(cMSE_{\ell}^{\eta_{\ell-1}}) = \frac{1}{M} \sum_{m=1}^M cMSE_{\ell,m}^{\eta_{\ell-1}}$$

10. Choose $\eta_{\ell-1}^*$.

$$\eta_{\ell-1}^* = \underset{\eta_{\ell-1}}{\operatorname{argmin}} E(cMSE_{\ell}^{\eta_{\ell-1}})$$

Then

$$\text{oDPLM}_{\ell-1,m} = \text{oDPLM}_{\ell,m}^{\eta_{\ell-1}^*},$$

$$\text{DPLM}_{\ell,m} = \text{DPLM}_{\ell,m}^{\eta_{\ell-1}^*},$$

$$E(\text{cMSE}_\ell) = E(\text{cMSE}_\ell^{\eta_{\ell-1}^*}).$$

11. If $E(\text{cMSE}_\ell) \geq E(\text{cMSE}_{\ell-1})$, then report $\ell - 1$ as the layer of prediction.

Otherwise, set $\ell = \ell + 1$ and repeat the step 8 through 10.

12. Construct the final MDPLM of $(\ell + 1)$ layers with \mathbf{D} .

6.2 Discussion

In this section, we will discuss the properties of MDPLM in two points of view, model assumption and bias-variance tradeoff.

6.2.1 Heterogeneity and Model Assumptions in MDPLM

To make a prediction model with heterogeneous data is difficult in that there are many uncertainties to resolve such as the covariate-response relationship, the form of variance component, the degree of heterogeneity, the choice of random effects, the interactions between covariates, etc. In general, solving such uncertainties requires a bunch of labor intensive works: plotting and a various statistic tests. Even after such labor intensive works, we cannot fixate such uncertainties in most cases because various combinations of heterogeneity and model complexity are possible (Figure 6.2.1). For instance, suppose that a dataset was generated from a mixture of lines which have similar slopes, not the same, and

pass through the origin. If we assume that the degree of heterogeneity is 1, target function would be linear with a covariate-dependent variance term. On the other hand, if we assume that target function is linear with an *i.i.d.* normal variance term, the degree of heterogeneity would be more than one and one of covariate would be a random effect. It is also possible that the random effect is an interaction of covariates, which is not directly observable. Although those model assumptions may fit the dataset well, it is not guaranteed that model assumption will be valid to unseen data.

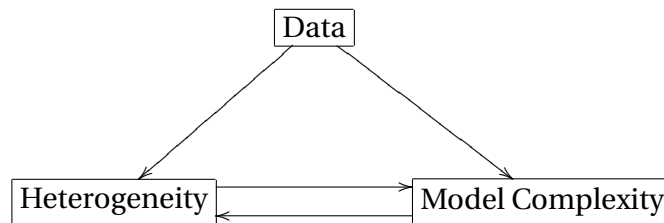


Figure 6.2.1: Relation of data, heterogeneity, and model complexity.

General Approaches for Prediction Modeling

Generally a prediction modeling requires model complexity and the degree of heterogeneity to be determined before training or parameter estimation (Figure 6.2.2). Among modeling approaches, which can deal with heterogeneous data, a linear (or nonlinear) mixed model and a generalized linear mixed model are categorized in this case. As mentioned earlier the decision about model complexity and heterogeneity cannot be made for heterogeneous data in most cases even after lots of works and statistic tests. *Multilayer perceptrons* (MLP), so-called artificial neural networks, can be also categorized in this group, but with little difference. Model complexity and heterogeneity are decided by the number of hidden layers,

the number of units in a hidden layer, and the form of sigmoid function (or activation function), which are fixed during training. Unlike mixed models, MLP does not require the form of target function and the form of variance terms. Since it is known that a two-hidden-layer perceptron provides full generality [39], heterogeneity, which MLP can deal with, is controlled by the number of units in a hidden layer when a hidden-unit activation function is fixed. If the number of hidden-layer units is set too low, the model will have high training and validation errors because of model underfitting and high statistical bias. On the other hand, with too many hidden-layer units, the model will have low training error, but have high validation error, due to model overfitting and high variance [40]. However, the optimal decision for the number of hidden-layer units is not straight-forward since it depends in a complicated way on the number of units in input and output layers, the size of training data, the form of error terms, the degree of heterogeneity, the type of activation functions, etc. Another important factor in MLP is the number of training iterations. Once the other factors, mentioned above, are fixed, early-termination of training has a risk of underfitting and too-many iterations may result in overfitting. Even though there are rule-of-thumbs in deciding those factors, it is hardly known how they are interacted.

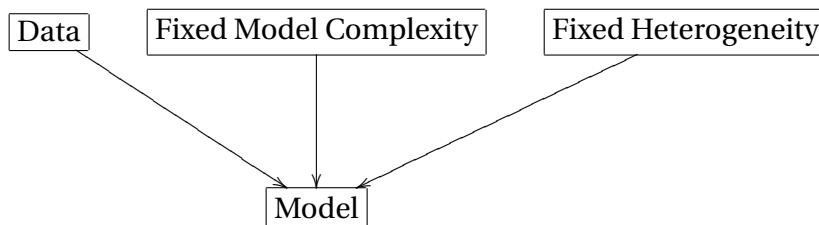


Figure 6.2.2: A general approach for modeling.

Generative Approaches for Prediction Modeling

A generative model is a probabilistic model which assumes that data observations are generated from a distribution parameterized by hidden variables. With parameter priors, the relationship between those hidden variables (or parameters) are represented by probability distributions and, also, the relationship between parameters and data is well formulated by posterior parameter distributions. *Dirichlet process* (DP) prior models such as DPLM, DPGLM, and MDPLM are categorized in a generative model.

A DP prior model is often called an infinite mixture model in that Dirichlet distribution, applied to the parameter space, constructs the dynamic parameter simplex. It means by the dynamic parameter simplex that the size of statistically different parameters can increase or decrease without upper-bound according to data distribution. In a general approach, the degree of heterogeneity is assumed to be known by specifying random effects (except MLP, the degree of heterogeneity is controlled by the number of hidden layer units in MLP). If the degree of heterogeneity is set too low due to a wrong selection of random effects, a model will suffer from underfitting and high statistical bias. On the contrary, if the degree of heterogeneity is erroneously set too high, the model will have the problem of overfitting and high variance (MLP also suffers the same problem due to a wrong decision on the number of hidden-layer units, which has been explained already in the previous section). Therefore, the decision on the degree of heterogeneity is critical to obtain a model with robustness. Unlike a general approach, a DP prior model approach does not require the degree of heterogeneity as a modeling input, that is, it is not necessary to specify which variable is a random effect to generate heterogeneity. Such a decision is made during parameter sampling according to the distribution of data under model assumptions.

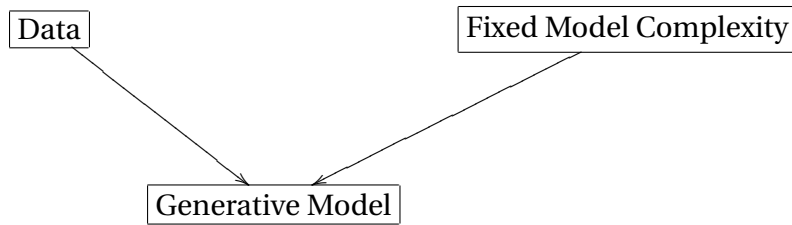


Figure 6.2.3: A generative probabilistic approach for modeling.

MDPLM as a Self-Adjusting Probabilistic Approach

As generative probabilistic models, DPLM and DPGLM do not have an upper-bound of the degree of heterogeneity. Under model assumptions with prior distributions of parameters, a parameter representing the degree of heterogeneity is decided by maximizing its posterior distribution. Therefore, the degree of heterogeneity still depends on a model assumption. When a model assumption is violated (e.g., a nonlinear function on a linear assumption in Section 4.2), the model will suffer from underfitting. It might be thought that MDPLM has the same assumption as DPLM. However, it is not true. Precisely speaking, a layer of MDPLM has the same assumption as DPLM. In MDPLM, the violation of assumption is sequentially remedied by layer generation equipped with covariate extension (Figure 6.2.4). Since the increase of layers may result in overfitting due to high variance, MDPLM selects a layer for prediction by balancing bias and variance.

6.2.2 Cross-validation and Degree of Heterogeneity in MDPLM

This section will explain the problems of cross-validation in heterogeneous data and discuss how MDPLM deals with the problematic situations in cross-validation.

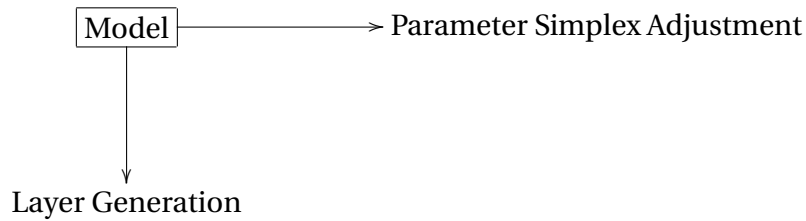


Figure 6.2.4: MDPLM as a self-adjusting probabilistic approach for modeling.

Bias-Variance Tradeoff

The bias-variance tradeoff is an important issue in every prediction model. The mean square error (or prediction error) of a model in expectation is decomposed into the variance of noise, the square of bias, and the variance of estimate (Eq. 6.2.1). The best model is one to minimize $E(MSE)$, equivalently, $E[(t_i - y_i)^2]$. In order to minimize $E(MSE)$, we need to minimize both the bias and the variance terms. However, it is very difficult and sometimes impossible because the real function is not known. In addition to uncertainty about a real function, the bias and the variance are interrelated. In an extreme example, we can make a model which always produce constant estimates, ignoring data. In that case, the variance term is zero, but the bias will be very high. On the other hand, if we had a model which perfectly estimates observations, that is, $y_i = t_i, \forall i$. This will make the bias term vanish (Eq. 6.2.2). However, the variance term will be the variance of error (Eq. 6.2.3). When the variance of noise is high, the variance of estimates will be significantly large. This is the usual problem of overfitting models. MLP is famous for its generality in that it can model any kinds of data, but also notorious for overfitting. When a model overfits a given dataset, the model learns the distribution of errors and loses robustness to unseen data.

$$E(MSE) = \frac{1}{N} \sum_{i=1}^N E[(t_i - y_i)^2]$$

$$E[(t_i - y_i)^2] = \underbrace{E[(t_i - f_i)^2]}_{Var(noise)} + \underbrace{\{f_i - E(y_i)\}^2}_{bias^2} + \underbrace{E[\{y_i - E(y_i)\}^2]}_{Var(y_i)} \quad (6.2.1)$$

where f_i is a real function value, $t_i (= f_i + \varepsilon_i)$ an observation for f_i , y_i an estimate for f_i by a model, and ε_i is *i.i.d.* with $E(\varepsilon_i) = 0$.

$$\text{When } y_i = t_i, \forall i, \quad E(y_i) = E(t_i) = E(f_i + \varepsilon_i) = f_i, \quad (6.2.2)$$

$$\begin{aligned} Var(y_i) &= E[\{y_i - E(y_i)\}^2] \quad (6.2.3) \\ &= E[(y_i - f_i)^2] \\ &= E[(t_i - f_i)^2] \\ &= E(\varepsilon_i^2) \\ &= Var(noise). \end{aligned}$$

Cross-Validation in Heterogeneous Data

Since the bias cannot be estimated directly, the cross-validation (CV) is frequently used to balance bias and variance. As for how to sample training-validation dataset and how to estimate the performance of a model, there exist several variants of CV such as resubstitution validation, hold-out validation, M -fold cross-validation, leave-one-out cross-validation, repeated M -fold cross-validation, etc. Among the variants, 10-fold CV is known to generate less biased estimation of error [41]. However, even 10-fold CV has some problems with heterogeneous data as well as other CVs. When CV is used to compare with models for their

performance, their (averaged) validation errors are used. The assumption of CV is that a validation dataset and a training dataset follow the distribution of data even after sampling. If this assumption holds, validation error will approach training error and the balance between bias and variance will maintain. In this case, the comparison of validation errors will be good for selecting a model and the estimated error will be the true performance of a model. Unfortunately, this ideal situation is rarely encountered and the general scenario is that validation error is much larger than training error. A common reason for this difference is wrong assumptions of model. As mentioned in Section 6.2.1, the general approach has strict assumptions and the violation of assumptions will result in model failure (inconsistency between model or data assumption and a real data distribution). However, the model assumption issue is not directly related to CV. Another reason of our interest is the way of sampling to make training-validation dataset pairs. Like modeling a classifier, a validation set for a model should be well stratified such that the set contains enough data necessary to extract the properties of the whole data, such as covariate-response relationships, the forms of error distribution, etc. However, different from classification data where response is discrete, we do not know how many different relations exist in data. Moreover, the number of classes (or the degree of heterogeneity) depends on model assumptions (see Section 6.2.1). Therefore, a validation dataset, sampled from the entire data without any strategy or algorithm, may not contain any information for a certain class and this will result in biased error (or performance) measure usually less than the true error. On the contrary, a validation set may take all information for a class, but make a training set empty for a class. In this case, a model will not have a chance to learn the class and, hence, will be underfitted with high bias.

Cross-Validation in MDPLM

In the previous section, we discussed the problems of cross-validation when the assumption about distributions of validation and training datasets is violated, which occurs frequently in heterogeneous data due to uncertainty of data properties. Another problem from the same reason is that validation error measure of CV does not balance the bias and the variance terms of $E(MSE)$ when the same distribution assumption is violated. For instance, if two models have the same validation error, CV does not discern one from another basically. Of course, a simple way is to take a model of the lower training error. However, there might be a case that a model has low training and high validation errors and another model has high training and low validation errors, but those errors are very close. CV will take the second model as a better one for the preference to low validation error. However, the low validation error of the second model might be biased due to an indiscreet sampling. This problematic situation may occur when both the size of validation data and the number of training-validation pairs are small (the variance of MSE is large).

MDPLM uses a different performance measure $E(cMSE)$, defined in Eq. 3.5.1, to balance training and validation errors. The error measure $cMSE$ utilizes entropy to resolve uncertainty from the difference of mean square errors between validation and training. The basic idea of $cMSE$ is to give more weight to an estimate of certainty. When $E(MSE_t)$ and $E(MSE_v)$ are the same, $E(cMSE)$ will have the same value and the performance measure will be less biased. When $E(MSE_t)$ and $E(MSE_v)$ are significantly different, $E(cMSE)$ will follow a higher value because those values might be biased and/or training-validation dataset pairs might be improperly sampled. The $cMSE$ ranges between MSE_t and MSE_v , inclusive, and gets close to the bigger value when the difference of two values increases. Therefore, the performance measure $E(cMSE)$ prevents an over- or underfitted model from being selected in model

selection by balancing training and validation errors.

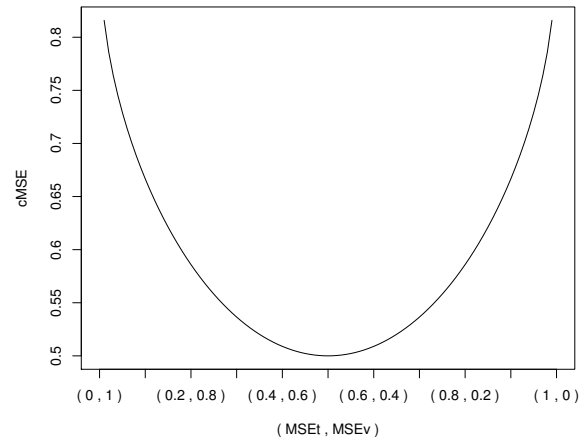


Figure 6.2.5: The error measure $cMSE(MSE_t, MSE_v)$ when $MSE_t, MSE_v \in [0, 1]$ and $MSE_v = 1 - MSE_t$ where MSE_t and MSE_v are the mean square errors of training and validation data, respectively.

Besides sampling (or generating from distributions) parameters of DPLMs and oDPLMs on layers, there are two kinds of decisions to construct an MDPLM: the sequence of designed noise rates and the layer of prediction. Such decisions are made, based on the performance measure $E(cMSE)$ conditional on possible values. In order to estimate $E(cMSE)$, CV is used with some modification in sampling. Although $E(cMSE)$ helps obtain a less biased performance measure, it does not contribute to sampling a stratified training-validation dataset pair. In order to generate a better dataset pair, we need to resolve uncertainties of data. However, we cannot fixate uncertainty because it depends on the assumption of a model (see Section 6.2.1). Reversely, if the assumption of a model is decided no matter whether it is correct or not, we can extract some information about data. We estimate the degree of heterogeneity by assuming that data have a linear mixture distribution. From a DPLM,

trained by the whole data, we can estimate the expected value of classes, denoted by $E(K)$. Assuming that each class takes the same proportion of data, each class will have $\frac{N}{E(K)}$ observations approximately. To make a training set have at least half of instances for each mixture component, the size of a validation set is set to $\frac{N}{2E(K)}$. In worst case that a validation takes all observations from a mixture component, its pairwise training set can still hold the rest half from the mixture component. Of course, this case may result in a biased performance estimate, but the strategy is that the sampling provides better training datasets, and the performance measure $cMSE$ discerns biased decisions from others so that biased decisions should not be chosen for the designed noise rate and the layer of prediction.

6.3 Conclusions

The proposed MDPLM is the integration of a generative probabilistic model and a performance estimation procedure to model complicated heterogeneous data without data analysis (or mining) and experts' knowledge in a field. MDPLM introduces the application of overfitting to improve a model by finding a hidden random effect. By combining layers of simple assumptions, MDPLM can deal with general complex situations. Moreover, MDPLM does not require any statistical test and human efforts to resolve uncertainties about data and, hence, can shorten the time of developing a regression model.

The first contribution of this dissertation is that we have developed an automated way of modeling heterogeneous data. A general way of modeling starts from data analysis and makes assumptions about data. Such a process often requires deep understanding of a field, related to data, as well as statistical knowledge. Especially, in heterogeneous data, lots of uncertainties are involved and such uncertainties are even interconnected. Therefore, traditional approaches, based on a fixed model assumption based on visualizing data and

statistical tests, have limitations to resolve uncertainties of complex heterogeneous data. As a generative approach, MDPLM fits itself to data. MDPLM automatically finds hidden random effects and increases model complexity by generating random effects and layers.

The second contribution is that we have developed a way of utilizing overfitting. Overfitting has been considered as the result of model failure. Based on the stochastic resonance theory, we found the usefulness of overfit models. MDPLM generates an overfit model, denoted by oDPLM, by adding noise and finds the direction of improvement by using the overfitting property (less bias).

The third contribution is that we have introduced a modified CV for better error estimate and model selection. Other CVs have a limitation that they are primarily focused on validation error under the assumption that models being compared have similar training errors. This results in a biased error estimate and wrong model selection when the assumption is violated. Our CV avoids such problematic situations by using an entropy-based error measure. In addition, we provided a sampling method in heterogeneous data to make training-validation dataset pairs. In result, a model, selected by our CV, will be better in the view of bias-variance tradeoff.

The fourth contribution is the finding that our performance measure $cMSE$ has interesting properties on the change of designed noise error and layer generation. The measure $cMSE$ is non-increasing to a minimal value as the number of layers increases. In contrast, $cMSE$ on the change of designed noise rate does not change much except points which occur with some frequency. Due to the finding, we can reduce the search space of the path for designed noise rates and the layer of prediction and can efficiently solve a multi-stage problem for the optimal layer of prediction.

The fifth contribution is concerning of the flexibility of the developed model. Unlike other regression models with fixed data (or model) assumptions, MDPLM has no assumption

about data, but has assumptions on its layers. The violation of layer assumption, caused by complex data, is alleviated by generating layers. As a self-adjusting model, MDPLM has two directions of generation, horizontally and vertically. Horizontally, with the Dirichlet process prior, the set of parameters is adjusted in size under the fixed data assumption. Besides, with the covariate extension, the number of layers increases according to data complexity. In result, MDPLM can fit various types of data and even the combination of different types of data.

6.4 Future Research

In this chapter, we will discuss possible future studies in order to make MDPLM more competitive against other existing regression models. MDPLM still has room for improvement as for parameter estimation, stratified sampling, and designed noise rate. Each section will discuss one of them and provide a possible solution.

Parameter estimation

A regression technique is often used to estimate parameters. It means by knowing parameters that we can formulate a response with covariates under a model assumption. The current implementation of MDPLM does not have the functionality of parameter estimation and, thus, a response is directly estimated from samples of parameters. However, once parameters are estimated, a relationship function between covariate and response will be available. For instance, suppose that we fitted data with MDPLM and that layer 1 turned out to be the layer of prediction. Then the response estimate function of the model will be of form " $exp(linear + \arctan(exp \times linear)) \times (linear + \arctan(exp \times linear))$ " as shown in Eq. 6.4.1.

$$\begin{aligned}
\text{layer 0: } y(x) &= \underbrace{\sum_{i=1}^{K^{(0)}} F_{x,i}^{(0)}(x) F_{y,i}^{(0)}(x)}_{\text{exp} \times \text{linear}} + e_0(x) \\
\text{layer 1: } e_0(x) &= \underbrace{\sum_{i=1}^{K^{(1)}} F_{x,i}^{(1)}(x, x_e(x)) F_{y,i}^{(2)}(x, x_e(x))}_{\text{exp}(\text{linear} + \arctan(\text{exp} \times \text{linear})) \times (\text{linear} + \arctan(\text{exp} \times \text{linear}))} + \varepsilon \quad (6.4.1) \\
x_e(x) &= \underbrace{g(\hat{y}(x) - \hat{y}'(x))}_{\arctan(\text{exp} \times \text{linear})}
\end{aligned}$$

where $F_{x,i}^{(j)}$ and $F_{y,i}^{(j)}$ are a covariate function and a covariate-response function, respectively, for mixture component i of layer j ; $K^{(j)}$ denotes the degree of heterogeneity of layer j ; \hat{y} and \hat{y}' are the response estimate functions at layer 0 by DPLM and oDPLM; ε denotes *i.i.d.* error of layer 1; x_e denotes the random effect function derived from layer 0; $e_0(x)$ is the error function of layer 0.

Most of models, based on MCMC samplers for parameters, have a statistical property called *label-switching*, which is equivalent to the lack of identifiability of mixture components [27]. Simply, if an integer k is sampled (or realized) for the degree of heterogeneity at an iteration, there will be $k!$ permutations which have the same likelihood. The loss-based parameter inference [23] is known to work well even though it is computationally expensive. However, this will not be directly applicable to our model because the complexity will exponentially increase by the increase of layers. Assume that each iteration has k mixture components. When there are T iterations in each layer and the layer of prediction is L (i.e., there are $L + 1$ layers), then there will be $(k!)^{T(L+1)}$ cases to consider and, thus, it is necessary to reduce the search space somehow.

Stratified sampling

As discussed earlier, when a given dataset is herogeneous, the stratified sampling is important in CV to obtain a less biased value for performance (or error) measure. Currently, MDPLM focuses on the goodness of training sets such that every training set can contain approximately at least half of observations from each mixture component. However, there may a case that a validation set will have observations from only one class, which will result in a biased performance measure. Therefore, we need a more sophisticated stratified sampler for both training and validation. A more reasonable way is to reduce the chance of event that validation datasets do not contain any observation from a class.

Let v denote the size of a validation set. Then the probability that the first validation set does not contain any observation from a class c is

$$\prod_{i=1}^v \left(1 - \frac{N/E(K)}{N-i} \right). \quad (6.4.2)$$

Similarly, assuming that the first validation set contains $n_{1,c}$ observations of the class c , the conditional probability that the second validation set does not contain any observation from the class c is

$$\prod_{i=1}^v \left(1 - \frac{N/E(K) - n_{1,c}}{N-v-i} \right) \frac{\binom{N/E(K)}{n_{1,c}}}{\binom{N}{v}}. \quad (6.4.3)$$

Then, the general term for validation set m is given by

$$\prod_{i=1}^v \left(1 - \frac{N/E(K) - \sum_{j=1}^{m-1} n_{j,c}}{N - (m-1)v - i} \right) \frac{\binom{N/E(K)}{\sum_{j=1}^{m-1} n_{j,c}}}{\binom{N}{(m-1)v}}. \quad (6.4.4)$$

Suppose that there are M validation datasets to collect. Let $n_{m,c}$ denote the number of observations from the class c in a validation set of index m . Since our purpose is to avoid any $n_{j,c}$ being zero, we can assume without loss of generality that every validation set contains at least one observation from the class c (i.e., $n_{j,c} \geq 1, \forall j$). In this case, every event described by Eq. 6.4.4 is mutually exclusive to each other and the probability that at most one of validation sets does not contain an observation from the class c is represented by

$$P = \sum_{m=1}^M \prod_{i=1}^v \left(1 - \frac{N/E(K) - \sum_{j=1}^{m-1} n_{j,c}}{N - (m-1)v - i} \right) \frac{\binom{N/E(K)}{\sum_{j=1}^{m-1} n_{j,c}}}{\binom{N}{(m-1)v}}. \quad (6.4.5)$$

Now, by taking expectation over $n_{j,c}$, we obtain the size of validation sets by minimizing the expectation as follows.

$$\begin{aligned} & \min_v E_{n_{1,c}, \dots, n_{M,c}}(P) & (6.4.6) \\ \text{s.t.} & \sum_{j=1}^M n_{j,c} \leq \frac{N}{E(K)} \\ & 1 \leq n_{j,c} \leq v, \forall j. \end{aligned}$$

Eq. 6.4.5 can be more simplified assuming that every class has the same number of data from the class c (i.e., $n_{j,c} = \alpha v, \forall j$, where $0 < \alpha < 1$).

Designed noise rate

Unlike the effect of layer transition on $E(cMSE)$, the designed noise rate has a more or less unpredictable effect on $E(cMSE)$. The current strategy of MDPLM for the designed noise rate is to search the discretized range upper-bounded to a prefixed value. Even if the upper-

bound can be lowered to a small value such as 0.1, computational complexity will be still large because the evaluation of $E(cMSE)$ entails training a model with a hidden random effect which is generated by a given designed noise rate. Thus, if there are M training-validation dataset pairs and S_η possible rates, there will be $2MS_\eta$ DPLMs to train and evaluate in a layer for CV. Another problem is about selecting an upper-bound of rates. The search range upper-bounded to 0.1 could be too wide. In case of Figure 3.5.1, the upper-bound of 0.05 will be good enough and will decrease the total modeling time approximately by a factor 2. Even though time complexity is linear of S_η , saving will be significant because of a huge amount time in training a DPLM. On the other hand, if the upper-bound is set to 0.03, most of noise rates will produce high $E(cMSE)$.

A possible improvement to reduce the search range will avoid the generation of rates of high $E(cMSE)$. Figure 3.5.1 shows that the noise rates of high $E(cMSE)$ occur intermittently. We suspect that such intermittent peculiar rates come from the interaction of noises introduced. If we assume that the effect of a noise, related to an observation, is a periodic function with its own frequency and that such a function has a peak in a period, then the occurrence of high $E(cMSE)$ could be explained by frequency resonance. Under the assumption, we could reduce the occurrence of interactions by make a noise focused on a single mixture rather than an observation in the following scheme.

$$\text{Noise is introduced with probability } 1 - h(x) \tag{6.4.7}$$

$$p_{max}(x) = \frac{\max_{j=1:K} F_{x,j}(x)}{\sum_{j=1}^K F_{x,j}(x)}$$

$$p'_{max}(x) = \frac{\max_{j=1:(K+1)} F_{x,j}(x)}{\sum_{j=1}^{K+1} F_{x,j}(x)}$$

$$h(x) = \eta_t \frac{p_{max}(x)}{p'_{max}(x)}$$

Then we expect that noise introduction by Eq. 6.4.7 will help prevent high $E(cMSE)$ from occurring. Therefore, if $E(cMSE)$ still does not change much as shown in Section 3.5.5, then we will be able to reduce the search range more.

REFERENCES

- [1] C. R. Henderson, "Best linear unbiased estimation and prediction under a selection model," *Biometrics*, vol. 31, no. 2, pp. 423–447, 1975.
- [2] P. McCullagh and J. Nelder, *Generalized Linear Models*. London: Chapman and Hall, second ed., 1989.
- [3] C. E. McCulloch, S. R. Searle, and J. M. Neuhaus, *Generalized, Linear, and Mixed Models (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2 ed., June 2008.
- [4] H. Goldstein, "Multilevel mixed linear model analysis using iterative generalized least squares," *Biometrika*, vol. 73, no. 1, pp. 43–56, 1986.
- [5] H. Goldstein, "Restricted unbiased iterative generalized least-squares estimation," *Biometrika*, vol. 76, pp. 622–623, September 1989.
- [6] H. Goldstein, "Nonlinear multilevel models, with an application to discrete response data," *Biometrika*, vol. 78, no. 1, pp. 45–51, 1991.
- [7] L. A. Hannah, D. M. Blei, and W. B. Powell, "Dirichlet process mixtures of generalized linear models," Jul 2010.
- [8] S. Mukhopadhyay and A. E. Gelfand, "Dirichlet process mixed generalized linear models," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 633–639, 1997.
- [9] B. Shahbaba and R. Neal, "Nonlinear models using dirichlet process mixtures," *J. Mach. Learn. Res.*, vol. 10, pp. 1829–1850, 2009.
- [10] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.
- [11] J. von Neumann, "Various techniques used in connection with random digits," *National Bureau of Standards, Applied Math Series*, vol. 11, pp. 36–38, 1951.
- [12] W. R. Gilks and P. Wild, "Adaptive rejection sampling for gibbs sampling," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 41, no. 2, pp. 337–348, 1992.
- [13] D. Spiegelhalter, A. Thomas, N. Best, and W. Gilks, "Bugs - bayesian inference using gibbs sampling version 0.50," tech. rep., 1995.
- [14] M. Plummer, "Jags: A program for analysis of bayesian graphical models using gibbs sampling," 2003.

- [15] J. Sethuraman, “A constructive definition of dirichlet priors,” *Statistica Sinica*, vol. 4, pp. 639–650, 1994.
- [16] D. Blackwell and J. B. MacQueen, “Ferguson distributions via polya urn schemes,” *The Annals of Statistics*, vol. 1, no. 2, pp. 353–355, 1973.
- [17] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, pp. 249–265, 2000.
- [18] M. D. Escobar and M. West, “Bayesian density estimation and inference using mixtures,” *Journal of the American Statistical Association*, vol. 90, pp. 577–588, 1994.
- [19] S. N. Maceachern and P. Müller, “Estimating mixture of dirichlet process models,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 2, pp. 223–238, 1998.
- [20] C. E. Antoniak, “Mixtures of dirichlet processes with applications to bayesian nonparametric problems,” *The Annals of Statistics*, vol. 2, pp. 1152–1174, November 1974.
- [21] S. Richardson and P. J. Green, “On bayesian analysis of mixtures with unknown number of components (with discussion),” *Journal of the Royal Statistical Society, Series B*, no. 59, pp. 731–792, 1997.
- [22] C. E. Rasmussen, “The infinite gaussian mixture model,” in *In Advances in Neural Information Processing Systems 12*, pp. 554–560, MIT Press, 2000.
- [23] M. Hurn, A. Justel, C. P. Robert, and C. P. Roberty, “Estimating mixtures of regressions,” 2000.
- [24] G. Casella and E. I. George, “Explaining the gibbs sampler,” *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.
- [25] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Hierarchical dirichlet processes,” *Journal of the American Statistical Association*, pp. 101–476, 2006.
- [26] G. Celeux, M. Hurn, and C. P. Robert, “Computational and inferential difficulties with mixture posterior distributions,” *Journal of the American Statistical Association*, no. 95, pp. 957–70, 2000.
- [27] M. Stephens, “Dealing with label switching in mixture models,” *Journal of the Royal Statistical Society, Ser. B*, no. 62, pp. 795–809, 2000.
- [28] H. Daume and D. Marcu, “A bayesian model for supervised clustering with the dirichlet process prior,” *Journal of Machine Learning Research*, no. 6, 2005.

- [29] P. Müller, G. L. Rosner, M. D. Iorio, and S. MacEachern, "A nonparametric bayesian model for inference in related longitudinal studies," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 54, no. 3, pp. 611–626, 2005.
- [30] R. W. M. Wedderburn, "Quasi-likelihood functions, generalized linear models, and the gauss-newton method," *Biometrika*, vol. 61, no. 3, pp. pp. 439–447, 1974.
- [31] H. Geidel, "Smirnow, n. w., und j. w. dunin-barkowski: Mathematische statistik in der technik. veb deutscher verlag der wissenschaften, berlin 1963," *Biometrische Zeitschrift*, vol. 7, no. 2, pp. 135–135, 1965.
- [32] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, "A theory of stochastic resonance in climatic change," *SIAM Journal on Applied Mathematics*, vol. 43, no. 3, pp. pp. 565–578, 1983.
- [33] B. McNamara and K. Wiesenfeld, "Theory of stochastic resonance," *Phys. Rev. A*, vol. 39, pp. 4854–4869, May 1989.
- [34] T. S. Breusch and A. R. Pagan, "A simple test for heteroscedasticity and random coefficient variation," *Econometrica*, vol. 47, no. 5, pp. pp. 1287–1294, 1979.
- [35] H. White, "A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity," *Econometrica*, vol. 48, no. 4, pp. pp. 817–838, 1980.
- [36] U.S. National Library of Medicine, "Prostate cancer." <http://www.nlm.nih.gov/medlineplus/ency/article/000380.htm>, March 2011.
- [37] M. Theiss, M. P. Wirth, A. Manseck, and H. G. W. Frohmüller, "Prognostic significance of capsular invasion and capsular penetration in patients with clinically localized prostate cancer undergoing radical prostatectomy," *The Prostate*, vol. 27, no. 1, pp. 13–17, 1995.
- [38] D. W. Hosmer and S. Lemeshow, *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, Sept. 2000.
- [39] E. D. Sontag, "Feedback stabilization using two-hidden-layer nets," *IEEE Trans. Neural Networks*, vol. 3, pp. 981–990, 1992.
- [40] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, January 1992.
- [41] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," pp. 1137–1143, Morgan Kaufmann, 1995.