KNOWLEDGE-BASED BAYESIAN LEARNING

A Dissertation

by

SHAHIN BOLUKI

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Edward R. Dougherty
Co-Chair of Committee,	Xiaoning Qian
Committee Members,	Krishna Narayanan
	Alan Dabney
Head of Department,	Miroslav M. Begovic

August 2021

Major Subject: Electrical Engineering

ABSTRACT

In engineering and life science applications, designing reliable and reproducible predictors is of utmost importance and interest. On one hand, the amount of available data for the application and problem of interest may be limited due to the costs associated with collecting or generating data in these domains. Limited relevant data can prohibit the effective design of such predictors. On the other hand, some form of *prior knowledge* is usually available even before observing any data, but is often neglected in predictor design. Bayesian approaches that are naturally equipped with uncertainty quantification are ideal candidates for these applications. In this dissertation, we develop methods and frameworks to leverage such prior knowledge, and data from other domains, if available, to improve the design of Bayesian predictors for the domain and application of interest.

We first propose a new prior construction methodology based on a general framework of constraints in the form of conditional probability statements. The new constraint framework is flexible as it naturally handles the potential inconsistency in archived relationships between the variables and conditioning can be augmented by other knowledge, such as population statistics. We demonstrate the effectiveness of our approach using pathway information and available knowledge of gene regulating functions for phenotypic classification. We then extend the method to mixture models which are useful in the presence of data heterogeneity.

Next, we focus on utilizing data from other domains to improve prediction accuracy in the target domain of interest. We develop a new generative model for optimal Bayesian supervised domain adaptation that can integrate next-generation sequencing data from different domains along with their labels, in addition to leveraging prior interactome knowledge. We show the superior performance of the proposed method, in terms of accuracy in identifying cancer subtypes by taking advantage of data from different domains and the available prior knowledge.

We then turn our attention to physical systems. First, we explain the concept of optimal experiment design under model uncertainty for autonomously collecting data and learning physical models. We discuss how prior construction fits in the overall design loop for an operator. We then

show how an efficient experiment design framework can accelerate exploration of the design space for a materials discovery application under model uncertainty.

Finally, we propose a novel framework of Bayesian reduced-order models for complex systems with high-dimensional systems dynamics or fields. In particular, we develop learnable Bayesian proper orthogonal decomposition that predicts the high-dimensional quantities of interest with reliable uncertainty estimates, in addition to embedding prior knowledge in terms of physics constraints. We showcase the proposed approach on predicting temperature and pressure fields.

DEDICATION

To my family, whose constant support helped me in this path.

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor, Dr. Edward Dougherty for his continuous support, guidance, and encouragement throughout my studies. I would also like to express my gratitude to my co-advisor, Dr. Xiaoning Qian for his invaluable collaboration, discussions, and support during my PhD. I would like to thank Dr. Krishna Narayanan and Dr. Alan Dabney for serving on my committee and their constructive suggestions.

Finally, I would like to especially thank my wife, my parents, and my brother, for their love, encouragement, and support throughout my life. Without them, I would not have come this far.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Edward Dougherty [advisor], Dr. Xiaoning Qian [co-advisor], and Dr. Krishna Narayanan of the Department of Electrical and Computer Engineering, and Dr. Alan Dabney of the Department of Statistics.

The data for the computational experiments in Chapter 6 was provided by collaborators from the Department of Materials Science and Engineering. The data used in the experiments of Chapter 7 was in part provided by collaborators from the Oden Institute for Computational Engineering and Sciences, University of Texas at Austin.

All work conducted for the dissertation was completed by the student, under the advisement of Dr. Edward Dougherty and Dr. Xiaoning Qian.

Funding Sources

Graduate study was in part supported by the National Science Foundation (NSF) Grants CCF-1553281, CMMI-1534534, and IIS-1812641.

NOMENCLATURE

OBC	Optimal Bayesian Classifier
MKDIP	Maximal Knowledge-Driven Information Prior
MCMC	Markov Chain Monte Carlo
REMLP	Regularized Expected Mean Log-Likelihood Prior
MMSE	Minimum Mean Square Error
RNA-Seq	RNA Sequencing
OBR	Optimal Bayesian Regression
GMM	Gaussian Mixture Model
EM	Expectation Maximization
OBSDA	Optimal Bayesian Supervised Domain Adaptation
NB	Negative Binomial
OBTD	Optimal Bayesian Classifier in the Target Domain
NN	Neural Network
NGS	Next Generation Sequencing
CRT	Chinese Restaurant Process
RLPP	Random Labeled Point Process
MOCU	Mean Objective Cost of Uncertainty
BMA	Bayesian Model Averaging
BED	Bayesian Experiment Design
BOMU	Bayesian Optimization under Model Uncertainty
BayPOD	Bayesian Proper Orthogonal Decomposition
ROM	Reduced Order Model

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGMENTS v

CONTRIBUTORS AND FUNDING SOURCES vi

NOMENCLATURE vii

TABLE OF CONTENTS viii

LIST OF FIGURES xii

LIST OF TABLES xvi

1. INTRODUCTION 1

2. INCORPORATING BIOLOGICAL PRIOR KNOWLEDGE FOR BAYESIAN LEARNING VIA MAXIMAL KNOWLEDGE-DRIVEN INFORMATION PRIORS 7

2.1 Introduction 7
2.2 Methods 10
2.2.1 Notation 10
2.2.2 Review of Optimal Bayesian Classification 11
2.2.3 Multinomial Mixture Model 13
2.2.4 Prior Construction: General Framework 14
2.2.5 Information-Theoretic Cost Functions 16
2.2.6 From Prior Knowledge to Mathematical Constraints 17
2.2.7 An Illustrative Example and Connection with Conditional Entropy 21
2.2.8 Special Case of Dirichlet Distribution 24
2.3 Results and Discussion 26
2.3.1 Mammalian Cell Cycle Classification 26
2.3.2 Classification Problem corresponding to TP53 29
2.3.3 Extracting General Constraints from Regulating Functions 30
2.3.4 Performance Comparison in Classification Setup 32
2.3.5 Performance Comparison in Mixture Setup 38
2.3.6 Performance Comparison on a Real Data Set 39

3. CONSTRUCTING PATHWAY-BASED PRIORS WITHIN A GAUSSIAN MIXTURE MODEL FOR BAYESIAN REGRESSION AND CLASSIFICATION 44

3.1 Introduction 44
3.2 Methods 45
3.2.1 Optimal Bayesian Regression and Classification for a Gaussian Mixture Model 45
3.2.1.1 Conjugate priors for Gaussian mixture model 49
3.2.2 Regularized Expected Mean Log-Likelihood Prior 50
3.2.2.1 Review of REMLP method for multivariate Gaussian with normal-Wishart prior distributions 53
3.2.3 Prior Construction and Inference for a GMM 55
3.2.3.1 Step 1: Initialization using Data 55
3.2.3.2 Step 2: Prior construction 57
3.2.3.3 Step 3: Prior update via Bayesian sampling 58
3.2.3.4 Step 4: Latent variable allocation and iteration 59
3.3 Results and Discussion 60
3.3.1 Simulation Setup 60
3.3.1.1 Synthetic pathway generation 60
3.3.1.2 Generating data from the synthetic pathways 63
3.3.1.3 Results 64
3.3.2 Performance on a Colon Cancer Pathway 72

4. OPTIMAL BAYESIAN SUPERVISED DOMAIN ADAPTATION FOR RNA SEQUENCING DATA 76

4.1 Introduction 76
4.2 Methods 79
4.2.1 OBSDA 79
4.2.2 SI-OBSDA 81
4.2.3 Incorporating Prior Network Knowledge in SI-OBSDA 84
4.2.4 Classification with OBSDA and SI-OBSDA 85
4.3 Results and Discussion 87
4.3.1 Data 87
4.3.2 Baselines 89
4.3.3 Results 90
4.3.3.1 LUAD and LUSC data in source and target domains 90
4.3.3.2 LUAD and LUSC data only in the target domain 92
4.3.3.3 Effect of incorporating prior knowledge 94

5. OPTIMAL CLUSTERING WITH MISSING VALUES 95

5.1 Introduction 95
5.2 Methods 97
5.2.1 Optimal Clustering 97
5.2.2 Gaussian Model with Missing Values 99

5.2.2.1 Gaussian means and known covariances 100
5.2.2.2 Gaussian-inverse-Wishart means and covariances 101
5.3 Results and Discussion 102
5.3.1 Simulated Data 103
5.3.2 RNA-seq Data 108

6. EXPERIMENT DESIGN UNDER MODEL UNCERTAINTY 111

6.1 Introduction 111
6.2 Generalized Mean Objective Cost of Uncertainty 112
6.2.1 Generalized MOCU 112
6.2.2 Connection of MOCU-based Experimental Design with KG and EGO 117
6.3 Efficient Experiment Design for Materials Discovery 119
6.3.1 Bayesian Optimization under Model Uncertainty 119
6.3.2 Building Robust Predictive Models through Bayesian Model Averaging 120
6.3.3 Experiment Design by Bayesian Optimization 122
6.3.4 Results and Discussion 125

7. BAYESIAN PROPER ORTHOGONAL DECOMPOSITION FOR LEARNABLE REDUCED-ORDER MODELS WITH UNCERTAINTY QUANTIFICATION 133

7.1 Introduction 133
7.2 Methods 137
7.2.1 Proper Orthogonal Decomposition (POD) 137
7.2.2 Physical Fields in the POD Basis 138
7.2.3 Learning POD Coefficients 138
7.2.4 Bayesian POD 139
7.2.5 BayPOD – A Generative POD Model 139
7.2.5.1 Inference model 140
7.2.5.2 Bayesian POD with linear mappings (BayPOD-LM) 143
7.2.5.3 Bayesian POD with neural networks (BayPOD-NN) 144
7.3 Results and Discussion 144
7.3.1 Heated Rod Example 144
7.3.1.1 Results 146
7.3.2 Airfoil Example 148
7.3.2.1 Results of BayPOD-LM and discussion 149
7.3.2.2 Results of BayPOD-NN and discussion 150

8. SUMMARY AND CONCLUSION 154

REFERENCES 157

APPENDIX A. ADDITIONAL RESULTS FOR CHAPTER 3 178

A.1 More Plots for the Results in Section 3.3.1.3 178
A.2 Single Component Regression Comparison Results 180

A.3 More Plots for the Results in Section 3.3.2 180

APPENDIX B. SUPPLEMENTARY MATERIALS FOR CHAPTER 4 187

B.1 OBSDA Inference via Gibbs Sampling 187
B.2 Joint Log-Likelihood of SI-OBSDA 190
B.3 Implementation Remarks for SI-OBSDA 190
B.4 A Note on the Difference Between the Proposed Model and Variational Autoencoders 191
B.5 Results on Subtyping of Endometrial Carcinoma 191

LIST OF FIGURES

FIGURE Page

2.1 A schematic illustration of the proposed Bayesian prior construction approach for a binary-classification problem. Information contained in the biological signaling pathways and their corresponding regulating functions is transformed to prior probabilities by MKDIP. Previously observed sample points (labeled or unlabeled) are used along with the constructed priors to design a Bayesian classifier to classify a new sample point (patient). 9

2.2 An illustrative example showing the components directly connected to gene 1. In the Boolean function $\{\text{AND, OR, NOT}\} = \{\wedge, \vee, -\}$. Based on the regulating function of gene 1, it is up-regulated if gene 5 is up-regulated and genes 2 and 3 are down-regulated. 21

2.3 Signaling pathways corresponding to Tables 2.1 and 2.2. Signaling pathways for: 2.3(a) the normal mammalian cell cycle (corresponding to Table 2.1) and 2.3(b) a simplified pathway involving TP53 (corresponding to Table 2.2) 28

2.4 Signaling pathways corresponding to NSCLC classification. The pathways are collected from KEGG Pathways for NSCLC and PI3K-AKT pathways, and from [1]. 41

3.1 A schematic for the prior construction method. 56

3.2 Toy example pathways. 57

3.3 Average regression and classification errors on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$ in the top and bottom panels respectively. 65

3.4 Average regression and classification errors on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$ in the top and bottom panels respectively. 66

3.5 Average component-conditional classification errors on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$ for the first and second components in the top and bottom panels respectively. 66

3.6 Average component-conditional classification errors on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$ for the first and second components in the top and bottom panels respectively. 67

3.7 Average F-score on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$. 67

3.8 Average F-score on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$. 68

3.9 Box plots of regression errors on synthetic pathways for different sample sizes with $p_1 = 0.6$ and $p_2 = 0.4$. 69

3.10 Box plots of classification errors on synthetic pathways for different sample sizes with $p_1 = 0.6$ and $p_2 = 0.4$. 70

3.11 A simplified colon-cancer-related pathway. 73

3.12 Performance on colon cancer pathways in Fig. 3.11. Average regression and classification errors with $\sigma_n^2 = 0.05$ in the top and bottom panels respectively. 73

3.13 Performance on colon cancer pathways in Fig. 3.11. Average regression and classification errors with $\sigma_n^2 = 0.1$ in the top and bottom panels respectively. 73

4.1 Schematic diagram of semi-implicit variational inference in SI-OBSDA 82

4.2 Average performance of different methods in identifying cancer subtypes of LUAD vs LUSC using different number of source samples. (t) and (t & s) correspond to using only target samples, and source and target samples in training, respectively. ... 91

5.1 Average clustering errors vs. missing probability for fixed means and covariances model. The first and second rows correspond to $n = 20$ and $n = 70$, respectively. ... 105

5.2 Average clustering errors vs. missing probability for Gaussian means and fixed covariances model. The first and second rows correspond to $n = 20$ and $n = 70$, respectively. 106

5.3 Average clustering errors for Gaussian means and inverse-Wishart covariances model. The first row corresponds to $n = 20$, and the errors are shown for different missing probabilities. The second row corresponds to $n = 70$ and missing probability of 0.15, where the errors are plotted vs. the Hamming distance threshold used to define the reference partitions in Pseed. 107

5.4 Empirical clustering errors on breast cancer RNA-seq data. 110

6.1 A design loop for designing optimal operators under uncertainty. 116

6.2 Schematic of the proposed framework for an autonomous, efficient materials discovery system as a realization of Bayesian Optimization under Model Uncertainty (BOMU). 126

6.3 Representative results for single objective optimization starting with 20 initial points using the best model F_2, worst model F_6, BMA_1 and BMA_2: a) Average maximum bulk modulus discovered, b) Average minimum shear modulus discovered 128

6.4 Average model probabilities for maximizing bulk modulus using a) BMA_1 and b) BMA_2 129

6.5 Representative results for single objective optimization – minimization of shear modulus for the case of 29 features: a) swarm plots indicating the distribution of the number of calculations required for convergence to the optimal solution using BMA_1 and F_{all} b) average model probabilities for maximizing bulk modulus using BMA_1 and F_{all}. 129

6.6 The Pareto optimal points in the materials property space are marked in red corresponding to the criterion of maximizing bulk modulus and minimizing shear modulus simultaneously. The Pareto set for the MDS consists of 10 points as indicated in red. 130

6.7 Average number of true Pareto optimal points found over all initial data set instances for single models, BMA_1, and BMA_2. 131

7.1 Schematic diagram of BayPOD at training and for prediction. Inputs can include settings for the parameters of the full (high-fidelity) model and initial or boundary conditions. 136

7.2 Four examples of comparing the actual temperature field and predictions from Polynomial Regression, BayPOD-LM, Neural Network Regression, and BayPOD-NN. 147

7.3 The minimum, mean, and maximum mean absolute error for (a) Polynomial Regression in top left, (b) BayPOD-LM in top right, (c) Neural Network Regression in bottom left, and (d) BayPOD-NN in bottom right. 150

7.4 The error field produced by predictions 151

7.5 The posterior predictive standard deviation from BayPOD-LM 151

7.6 The prediction from BayPOD-NN 152

A.1 Box plots of regression errors on synthetic pathways for different sample sizes with $p_1 = 0.72$ and $p_2 = 0.28$. 178

A.2 Box plots of classification errors on synthetic pathways for different sample sizes with $p_1 = 0.72$ and $p_2 = 0.28$. 179

A.3 Average regression error vs sample size in a single component problem. 181

A.4 Box plots of regression errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.05$. 182

A.5 Box plots of classification errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.05$. 183

A.6 Box plots of regression errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.1$. 184

A.7 Box plots of classification errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.1$. 185

A.8 Average component-conditional classification errors on colon cancer pathways with $\sigma_n^2 = 0.05$ for the first and second components in the top and bottom panels respectively. 185

A.9 Average component-conditional classification errors on colon cancer pathways with $\sigma_n^2 = 0.1$ for the first and second components in the top and bottom panels respectively. 186

A.10 Average F-score on colon cancer pathways with $\sigma_n^2 = 0.05$. 186

A.11 Average F-score on colon cancer pathways with $\sigma_n^2 = 0.1$. 186

LIST OF TABLES

TABLE Page

2.1 Boolean regulating functions of normal mammalian cell cycle adapted from [2]. In the Boolean functions $\{$AND, OR, NOT$\} = \{\wedge, \vee, -\}$. 27

2.2 Boolean regulating functions corresponding to the pathway in Figure 2.3(b) adapted from [3]. In the Boolean functions $\{$AND, OR, NOT$\} = \{\wedge, \vee, -\}$. 29

2.3 The set of constraints extracted from the regulating functions and pathways for the TP53 network. Constraints extracted from the Boolean regulating functions in Table 2.2 corresponding to the pathway in Figure 2.3(b) used in MKDIP-E, MKDIP-D, MKDIP-R (left). Constraints extracted from the pathway in Figure 2.3(b) used in RMEP, RMDIP, REMLP (right). 32

2.4 Expected true error of different classification rules for the mammalian cell-cycle network. The constructed priors are considered using two precision factors: optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$, where the minimum achievable error (Bayes error) is denoted by Err_{Bayes}. The lowest error for each sample size is written in bold. 35

2.5 Expected true error of different classification rules for the TP53 network. The constructed priors are considered using two precision factors: optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$, where the minimum achievable error (Bayes error) is denoted by Err_{Bayes}. The lowest error for each sample size is written in bold. 36

2.6 Expected difference between the true model (for mammalian cell-cycle network) and estimated posterior probability masses. Optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$. The lowest distance for each sample size is written in bold 37

2.7 Expected difference between the true model (for TP53 network) and estimated posterior probability masses. Optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$. The lowest distance for each sample size is written in bold. 37

2.8 Expected errors of different Bayesian classification rules in the mixture model for the mammalian cell-cycle network. Expected true error (left) and expected error on unlabeled training data (right), with $c_0 = 0.6$. The lowest error for each sample size and the lowest error among practical methods is written in bold. 39

2.9 Expected errors of different Bayesian classification rules in the mixture model for the TP53 network. Expected true error (left) and expected error on unlabeled training data (right), with $c_0 = 0.6$. The lowest error for each sample size and the lowest error among practical methods is written in bold. 40

2.10 Regulating functions corresponding to the signaling pathways in Figure 2.4. In the Boolean functions $\{\text{AND, OR, NOT}\} = \{\wedge, \vee, -\}$. 42

2.11 Expected error of different classification rules calculated on a real dataset. The classification is between LUA (class 0) and LUS (class 1), with $c = 0.57$. 43

3.1 Conjugate Prior for Gaussian Mixture 49

3.2 Input Parameters Used in Generating Pathways. 62

4.1 Average errors (in % $\pm$ standard deviations) in identifying subtypes of LUAD vs LUSC with the source domain containing samples from the same subtypes. 92

4.2 Average errors (in % $\pm$ standard deviations) in identifying subtypes of LUAD vs LUSC with the source domain containing samples from different labels. 93

4.3 Comparison of SI-OBSDA and SI-OBSDA without prior knowledge (SI-OBSDA w/o Prior) in terms of average errors (in %) in identifying subtypes of LUAD vs LUSC with different source domain settings. 94

5.1 Parameters for the point generation under three models. N, IW, $\mathbf{1}_d$, and I_d denote Gaussian, inverse-Wishart, column vector of all ones with length d, and $d \times d$ idendity matrix, respectively. 104

7.1 Mean, standard deviation, minimum, and maximum of mean absolute error for Polynomial Regression, BayPOD-LM, Neural Network Regression, and BayPOD-NN on the different test snapshots for the heated rod case study. 146

B.1 Average errors (in % $\pm$ standard deviations) and AUC ($\pm$ standard deviations) in identifying endometrioid endometrial vs serous endometrial with the source domain containing samples from ovarian serous adenocarcinoma. 192

1. INTRODUCTION

In many engineering and biological applications, the amount of available data is limited. On one hand, engineering applications often require running expensive (in terms of money and/or computation and time) real-world or simulation-based experiments to generate data. On the other hand, collecting appropriate data for biological applications, for example from complex diseases, is a costly procedure, if not prohibitive, considering the clinical, biological, and technical challenges involved in the process. Given the prevalent data heterogeneity in complex diseases like cancer [4], usually more samples are needed than what can be collected to achieve reliable predictors. These limitations can prohibit collecting enough samples for the problem of interest to design a reproducible predictor. In such circumstances, model-free classification, regression, or clustering may become virtually impossible.

Integrating the existing *prior knowledge* into the design of predictors and operators for these applications becomes an inevitable choice to improve both reliability and accuracy while maintaining interpretability in terms of agreeing with prior belief. Prior knowledge may have been compiled by combining experimental support from several relevant studies over the years. For example, the interactome knowledge can be a condensation of several different studies/databases including protein-protein and regulatory interactions, signaling interactions, metabolic pathway interactions, and kinase-substrate interactions. Or for physical systems, there exists extensive knowledge about physical constraints and/or relationships between physical properties governed by physics equation that can neither be ignored nor overruled by an extrapolating model trained on data. Clearly, machine learning models that only focus on the data at hand and do not leverage prior knowledge overlook a potential wealth of relevant information regarding the target task. Moreover, for many *target domains* that lack enough data for designing reliable predictors and operators, data from other domains exist which can prove helpful.

In this dissertation we address the aforementioned problems by developing frameworks for incorporating prior knowledge within Bayesian machine learning models, proposing a new Bayesian

method for supervised domain adaptation to utilize data from other domains with the capability of leveraging prior knowledge, and designing a novel Bayesian reduced-order modeling with uncertainty quantification that is faithful to the prior knowledge. The applications considered in Chapters 2 to 5 are life science related, and in Chapters 6 and 7 are concerning physical systems. In the following, we briefly discuss the problems considered in the different Chapters and the proposed solutions. More background and details can be found in each Chapter.

In Chapter 2 we develop a new framework for incorporating prior knowledge in classification and in Chapter 3 we extend it to Gaussian mixture models and regression. Phenotypic classification, biomarker estimation, and patient outcome prediction based on genomic data are among the most important current issues in translational genomics. All remain problematic because there are often tens of thousands of potential features with very small samples (either labeled or unlabeled), typically under 100. These problems become more challenging given the inherent inter- and intra-heterogeneity in tumor samples. In such circumstances, the use of prior knowledge becomes critical, where rather than depending only on expression data, one can use genetic pathway information to augment classifier or regressor design. We aim to incorporate knowledge in terms of genetic pathways, which have been compiled over several years and studies, in the machine learning process. Optimal Bayesian classification/regression concept provides optimal classification/regression under model uncertainty. It differs from classical Bayesian methods in which a classification/regression model is assumed and prior distributions are placed on model parameters. With optimal Bayesian classification/regression, uncertainty is treated directly on the feature-label/predictor-target distribution, which assures full utilization of prior knowledge and is guaranteed to outperform classical methods under the model assumptions. The salient problem confronting optimal Bayesian methods is prior construction, which becomes specially important when the available data contain smaller sample sizes (with respect to the number of features). In Chapter 2, we propose a new prior construction methodology based on a general framework of constraints in the form of conditional probability statements. We call this prior the maximal knowledge-driven information prior (MKDIP). The new constraint framework is flexible and can

naturally handle the potential inconsistency in archived regulatory relationships and conditioning can be augmented by other knowledge, such as population statistics. The performance of the proposed methods is examined on two important pathway families, the mammalian cell-cycle and a set of p53-related pathways, and also on a publicly available gene expression dataset of non-small cell lung cancer when combined with the existing prior knowledge on relevant signaling pathways. We demonstrate the effectiveness of our approach using pathway information and available knowledge of gene regulating functions; however, the underlying theory can be applied to a wide variety of knowledge types, and other applications when there are small samples. The applications in Chapter 2 contain discrete data. We extend the application of prior construction to Gaussian mixture models as well as regression problems in Chapter 3, which is useful in the presence of unknown labels or data heterogeneity. The performance is validated on phenotype classification and biomarker estimation when the prior knowledge consists of colon cancer pathways. In Chapters 2 and 3 we see that the proposed framework results in better inference when proper prior knowledge exists.

When learning to subtype complex disease based on next-generation sequencing data, the amount of available data is often limited. Recent works based on transfer learning and domain adaptation have tried to leverage data from other domains to design better predictors in the target domain of interest with varying degrees of success. But they are either limited to the cases requiring the outcome label correspondence across domains or cannot leverage the label information at all. Moreover, the existing methods cannot usually benefit from other information available a priori such as gene interaction networks. In Chapter 4, we develop a generative optimal Bayesian supervised domain adaptation (OBSDA) model that can integrate RNA sequencing (RNA-Seq) data from different domains along with their labels for improving prediction accuracy in the target domain. Our model can be applied in cases where different domains share the same labels or have different ones. OBSDA is based on a hierarchical Bayesian negative binomial model with parameter factorization, for which the optimal predictor can be derived by marginalization of likelihood over the posterior of the parameters. We first provide an efficient Gibbs sampler for parameter

inference in OBSDA. Then, we leverage the gene-gene network prior information and construct an *informed* and flexible variational family to infer the posterior distributions of model parameters. Comprehensive experiments on real-world RNA-Seq data demonstrate the superior performance of OBSDA, in terms of accuracy in identifying cancer subtypes by utilizing data from different domains. Moreover, we show that by taking advantage of the prior network information we can further improve the performance.

In Chapter 5, we focus on the problem of clustering in the presence of missing values and showcase our proposed method in biomedical studies. Missing values can complicate the application of clustering algorithms, whose goals are to group points based on some similarity criterion. In modern biomedical studies, missing values frequently arise due to various reasons, including missing tests or complex profiling technologies for different omics measurements. Clustering of expression profiles taken over various tissue samples is usually done with the aim of discriminating pathologies based on differential patterns of gene expression. A common practice for dealing with missing values in the context of clustering is to first impute the missing values, and then apply the clustering algorithm on the completed data, but this approach faces difficulties in small-sample settings. We consider missing values in the context of optimal clustering, which finds an optimal clustering operator with reference to an underlying random labeled point process (RLPP). We show how the missing-value problem fits neatly into the overall framework of optimal clustering by incorporating the missing value mechanism into the random labeled point process and then marginalizing out the missing-value process. While we do not utilize any specific prior knowledge in Chapter 5, we address the problem of clustering with missing values under smaller sample settings. Comprehensive experimental studies on both synthetic and real-world RNA-seq data show the superior performance of the proposed optimal clustering with missing values when compared to various clustering approaches, while obviating the need for imputation-based pre-processing of the data. Since we demonstrate the proposed framework for the Gaussian model with arbitrary covariance structures, the application is general and not limited to the studied area.

In Chapter 6, we first explain the concept of optimal experiment design and propose a general

utility function for guiding experiments. Optimal experiment design prioritizes experiments or actively collects data for autonomously learning models and reducing the uncertainty most pertinent to the operational cost/objective. Optimal experiment design is critical for applications where performing each experiment or collecting data is expensive (in terms of money, time, or resources). We demonstrate how this new formulation includes as special cases some of the widely used existing approaches, and discuss how prior construction fits in the overall design loop for an operator. We then develop an efficient experiment design framework under model uncertainty, where prior knowledge in terms of potential models or feature sets exist. Our framework is demonstrated on a materials discovery problem, by efficiently exploring the MAX ternary carbide/nitride space through density functional theory (DFT) calculations. Usually in experiment design problems, the goal is to start the experiment design loop as soon as possible (with the least amount of initial experiments/data) to use resources more efficiently. This can significantly prohibit reliable model selection. We see that the proposed framework is capable of autonomously and adaptively learning not only the most promising regions in the design space but also the models that most efficiently guide such exploration.

Finally, in Chapter 7, we develop a new framework of Bayesian reduced-order models. Appropriate mathematical modeling of systems dynamics is essential for designing and controlling complex systems in science and engineering. Recent works have explored the connection between reduced-order models of high-dimensional differential equation systems and surrogate machine learning models. However, their focus has been how to best approximate the high fidelity model of choice. We propose a novel framework of Bayesian reduced-order models naturally equipped with uncertainty quantification. In particular, we develop learnable Bayesian proper orthogonal decomposition (BayPOD) that learns the distributions of both the POD projection bases and the mapping from the system input parameters to the projected scores/coefficients so that the learned BayPOD can help predict high-dimensional systems dynamics/fields as quantities of interest in different setups with reliable uncertainty estimates. The developed learnable BayPOD has the capability of embedding physics constraints when learning the POD-based surrogate reduced-order

models, a desirable feature when studying complex systems in science and engineering applications where the available training data are limited. Furthermore, the proposed BayPOD method is an end-to-end solution, which unlike other surrogate-based methods, does not require separate POD and machine learning steps. The results from case studies of predicting the temperature field of a heated rod and the pressure field around an airfoil shows the potential of learnable BayPOD as a new family of reduced-order models with reliable uncertainty estimates.

2. INCORPORATING BIOLOGICAL PRIOR KNOWLEDGE FOR BAYESIAN LEARNING VIA MAXIMAL KNOWLEDGE-DRIVEN INFORMATION PRIORS *

2.1 Introduction

Small samples are commonplace in phenotypic classification and, for these, prior knowledge is critical [5, 6]. If knowledge concerning the feature-label distribution is available, say, genetic pathways, then it can be used to design an optimal Bayesian classifier (OBC) for which uncertainty is treated directly on the feature-label distribution. As typical with Bayesian methods, the salient obstacle confronting OBC is prior construction. In this Chapter, we propose a new prior construction framework to incorporate gene regulatory knowledge via general types of constraints in the form of probability statements quantifying the probabilities of gene up- and down-regulation conditioned on the regulatory status of other genes. We extend the application of prior construction to a multinomial mixture model when labels are unknown, a key issue confronting the use of data arising from unplanned experiments in practice.

Regarding prior construction, E. T. Jaynes has remarked [7], "... there must exist a general formal theory of determination of priors by logical analysis of prior information – and that to develop it is today the top priority research problem of Bayesian theory". It is precisely this kind of formal structure that is presented in this Chapter. The formal structure involves a constrained optimization in which the constraints incorporate existing scientific knowledge augmented by slackness variables. The constraints tighten the prior distribution in accordance with prior knowledge, while at the same time avoiding inadvertent over restriction of the prior, an important consideration with small samples.

Subsequent to the introduction of Jeffreys' non-informative prior [8], there was a series of information-theoretic and statistical methods: Maximal data information priors (MDIP) [9], non-informative priors for integers [10], entropic priors [11], reference (non-informative) priors ob-

*Reprinted with permission from S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Incorporating biological prior knowledge for Bayesian learning via maximal knowledge-driven information priors," BMC Bioinformatics, vol. 18, no. 14, pp. 61–80, 2017. Copyright 2017 Authors.

tained through maximization of the missing information [12], and least-informative priors [13] (see also [14, 15, 16] and the references therein). The principle of maximum entropy can be seen as a method of constructing least-informative priors [17, 18], though it was first introduced in statistical mechanics for assigning probabilities. Except in the Jeffreys' prior, almost all the methods are based on optimization: max- or min-imizing an objective function, usually an information theoretic one. The least-informative prior in [13] is found among a restricted set of distributions, where the feasible region is a set of convex combinations of certain types of distributions. In [19], several non-informative and informative priors for different problems are found. All of these methods emphasize the separation of prior knowledge and observed sample data.

Although the methods above are appropriate tools for generating prior probabilities, they are quite general methodologies without targeting any specific type of prior information. In that regard, the problem of prior selection, in any Bayesian paradigm, is usually treated conventionally (even "subjectively") and independent of the real available prior knowledge and sample data. Figure 2.1 shows a schematic view of the proposed mechanism for Bayesian operator design.

The *a priori* knowledge in the form of graphical models (e.g., Markov random fields) has been widely utilized in covariance matrix estimation in Gaussian graphical models. In these studies, using a given graphical model illustrating the interactions between variables, different problems have been addressed: e.g., constraints on the matrix structure [20, 21] or known independencies between variables [22, 23]. Nonetheless, these studies rely on a fundamental assumption: the given prior knowledge is complete and hence provides one single solution. However, in many applications including genomics, the given prior knowledge is uncertain, incomplete, and may be inconsistent. Therefore, instead of interpreting the prior knowledge as a single solution, e.g., a single deterministic covariance matrix, we aim at constructing a prior distribution on an uncertainty class.

In a different approach to prior knowledge, gene-gene relationships (pathway-based or protein-protein interaction (PPI) networks) are used to improve classification accuracy [24, 25, 26, 27, 28, 29, 30], consistency of biomarker discovery [31, 32], accuracy of identifying differentially

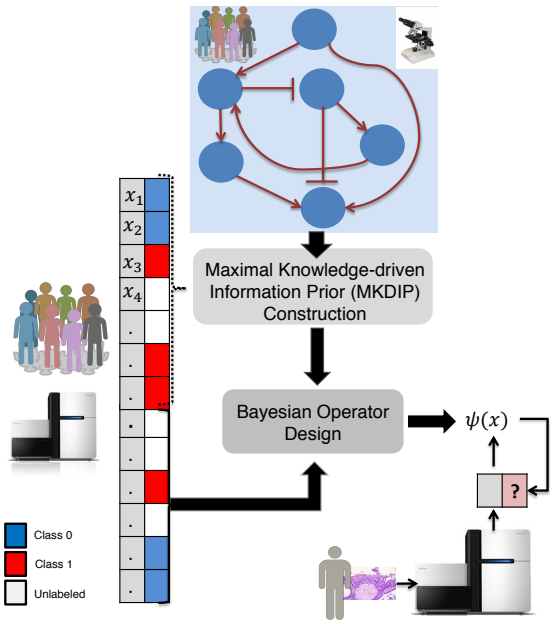


Figure 2.1: A schematic illustration of the proposed Bayesian prior construction approach for a binary-classification problem. Information contained in the biological signaling pathways and their corresponding regulating functions is transformed to prior probabilities by MKDIP. Previously observed sample points (labeled or unlabeled) are used along with the constructed priors to design a Bayesian classifier to classify a new sample point (patient).

expressed genes and regulatory target genes of a transcription factor [33, 34, 35], and targeted therapeutic strategies [36, 37]. The majority of these studies utilize gene expressions corresponding to sub-networks in PPI networks, for instance: mean or median of gene expression values in gene ontology network modules [24], probabilistic inference of pathway activity [28], and producing candidate sub-networks via a Markov clustering algorithm applied to high quality PPI networks [30, 38]. None of these methods incorporate the regulating mechanisms (activating or suppressing) into classification or feature-selection to the best of our knowledge.

The fundamental difference of the work presented in this Chapter is that we develop machinery to transform knowledge contained in biological signaling pathways to prior probabilities. We propose a general framework capable of incorporating any source of prior information by extending previous prior construction methods [39, 40]. We call the final prior distribution constructed via this framework, a *maximal knowledge-driven information prior* (MKDIP). The new MKDIP con-

struction constitutes two steps: (1) Pairwise and functional information quantification: information in the biological pathways is quantified by an information theoretic formulation. (2) Objective-based Prior Selection: combining sample data and prior knowledge, we build an objective function, in which the expected mean log-likelihood is regularized by the quantified information in step 1. As a special case, where we do not have any sample data, or there is only one data point available for constructing the prior probability, the proposed framework is reduced to a regularized extension of the maximum entropy principle (MaxEnt) [41].

Owing to population heterogeneity we often face a *mixture model*, for example, when considering tumor sample heterogeneity where the assignment of a sample to any subtype or stage is not necessarily given. Thus, we derive the MKDIP construction and OBC for a mixture model. In this Chapter, we assume that data are categorical, e.g. binary or ternary gene-expression representations. In the next Chapter, the case with continuous data is addressed. Such categorical representations have many potential applications, including those wherein we only have access to a coarse set of measurements, e.g. epifluorescent imaging [42], rather than fine-resolution measurements such as microarray or RNA-Seq data. Finally, we emphasize that, in our framework, no single model is selected; instead, we consider all possible models as the uncertainty class that can be representative of the available prior information and assign probabilities to each model via the constructed prior.

2.2 Methods

2.2.1 Notation

Boldface lower case letters represent column vectors. Occasionally, concatenation of several vectors is also shown by boldface lower case letters. For a vector $\boldsymbol{a}$, a_0 represents the summation of all the elements and a_i denotes its $i-$th element. Probability sample spaces are shown by calligraphic uppercase letters. Uppercase letters are for sets and random variables (vectors). Probability measure over the random variable (vector) X is denoted by $P(X)$, whether it be a probability density function or a probability mass function. $E_X[f(X)]$ represents the expectation

of $f(X)$ with respect to X. $P(\boldsymbol{x}|\boldsymbol{y})$ denotes the conditional probability $P(X = \boldsymbol{x}|Y = \boldsymbol{y})$. $\boldsymbol{\theta}$ represents generic parameters of a probability measure, for instance $P(X|Y;\boldsymbol{\theta})$ (or $P_{\boldsymbol{\theta}}(X|Y)$) is the conditional probability parameterized by $\boldsymbol{\theta}$. $\boldsymbol{\gamma}$ represents generic hyperparameter vectors. $\pi(\boldsymbol{\theta};\boldsymbol{\gamma})$ is the probability measure over the parameters $\boldsymbol{\theta}$ governed by hyperparameters $\boldsymbol{\gamma}$, the parameters themselves governing another probability measure over some random variables. $\mathcal{M}ult(\boldsymbol{p};n)$ and $\mathcal{D}(\boldsymbol{\alpha})$ represent a multinomial distribution with vector parameter $\boldsymbol{p}$ and n samples, and a Dirichlet distribution with vector $\boldsymbol{\alpha}$, respectively.

2.2.2 Review of Optimal Bayesian Classification

Binary classification involves a feature vector $\mathbf{X} = (X_1, X_2, ..., X_d)^T \in \Re^d$ composed of random variables (features), a binary random variable (label) Y and a *classifier* $\psi(\mathbf{X})$ to predict Y. The error is $\varepsilon[\psi] = P(\psi(\mathbf{X}) \neq Y)$. An optimal classifier, ψ_{bay}, called a *Bayes classifier*, has minimal error, called the *Bayes error*, among all possible classifiers. The underlying probability model for classification is the joint feature-label distribution. It determines the class prior probabilities $c_0 = c = P(Y = 0)$ and $c_1 = 1 - c = P(Y = 1)$, and the class-conditional densities $f_0(\mathbf{x}) = P(\mathbf{x}|Y = 0)$ and $f_1(\mathbf{x}) = P(\mathbf{x}|Y = 1)$. A Bayes classifier is given by

$$\psi_{\text{bay}}(\mathbf{x}) = \begin{cases} 1\,, & c_1 f_1(\mathbf{x}) \geq c_0 f_0(\mathbf{x})\,, \\ 0\,, & \text{otherwise.} \end{cases} . \tag{2.1}$$

If the feature-label distribution is unknown but belongs to an uncertainty class of feature-label distributions parameterized by the vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, then, given a random sample S_n, an *optimal Bayeisan classifier* (OBC) minimizes the expected error over $\boldsymbol{\Theta}$:

$$\psi_{\text{OBC}} = \arg\min_{\psi \in \mathcal{C}} E_{\pi^*(\theta)}[\varepsilon_{\boldsymbol{\theta}}[\psi]], \tag{2.2}$$

where the expectation is relative to the posterior distribution $\pi^*(\boldsymbol{\theta})$ over $\boldsymbol{\Theta}$, which is derived from the prior distribution $\pi(\boldsymbol{\theta})$ using Bayes' rule [43, 44]. If we let $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ denote the class 0

and class 1 parameters, then we can write $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = [c, \boldsymbol{\theta_0}, \boldsymbol{\theta_1}]$. If we assume that $c, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1$ are independent prior to observing the data, i.e. $\pi(\boldsymbol{\theta}) = \pi(c)\pi(\boldsymbol{\theta}_0)\pi(\boldsymbol{\theta}_1)$, then the independence is preserved in the posterior distribution $\pi^*(\boldsymbol{\theta}) = \pi^*(c)\pi^*(\boldsymbol{\theta}_0)\pi^*(\boldsymbol{\theta}_1)$ and the posteriors are given by $\pi^*(\boldsymbol{\theta}_y) \propto \pi(\boldsymbol{\theta}_y)\prod_{i=1}^{n_y} f_{\boldsymbol{\theta}_y}(\mathbf{x}_i^y|y)$ for $y = 0, 1$, where $f_{\boldsymbol{\theta}_y}(\mathbf{x}_i^y|y)$ and n_y are the class-conditional density and number of sample points for class y, respectively [45].

Given a classifier ψ_n designed from random sample S_n, from the perspective of mean-square error, the best error estimate minimizes the MSE between its true error (a function of $\boldsymbol{\theta}$ and ψ_n) and an error estimate (a function of S_n and ψ_n). This Bayesian minimum-mean-square-error (MMSE) estimate is given by the expected true error, $\widehat{\varepsilon}(\psi_n, S_n) = \mathrm{E}_{\boldsymbol{\theta}}[\varepsilon(\psi_n, \boldsymbol{\theta})|S_n]$, where $\varepsilon(\psi_n, \boldsymbol{\theta})$ is the error of ψ_n on the feature-label distribution parameterized by $\boldsymbol{\theta}$ and the expectation is taken relative to the prior distribution $\pi(\boldsymbol{\theta})$ [45]. The expectation given the sample is over the posterior probability. Thus, $\widehat{\varepsilon}(\psi_n, S_n) = \mathrm{E}_{\pi^*}[\varepsilon]$.

The *effective class-conditional density* for class y is defined by

$$f_{\boldsymbol{\Theta}}(\mathbf{x}|y) = \int_{\boldsymbol{\Theta}_y} f_{\boldsymbol{\theta}_y}(\mathbf{x}|y)\, \pi^*(\boldsymbol{\theta}_y)\, d\boldsymbol{\theta}_y, \tag{2.3}$$

$\boldsymbol{\Theta}_y$ being the space for $\boldsymbol{\theta}_y$, and an OBC is given pointwise by [43]

$$\psi_{\mathrm{OBC}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathrm{E}_{\pi^*}[c] f_{\boldsymbol{\Theta}}(\mathbf{x}|0) \geq (1 - \mathrm{E}_{\pi^*}[c]) f_{\boldsymbol{\Theta}}(\mathbf{x}|1), \\ 1 & \text{otherwise.} \end{cases} \tag{2.4}$$

For discrete classification there is no loss in generality in assuming a single feature X taking values in the set $\{1, \ldots, b\}$ of "bins". Classification is determined by the class 0 prior probability c and the class-conditional probability mass functions $p_i = P(X = i|Y = 0)$ and $q_i = P(X = i|Y = 1)$, for $i = 1, \ldots, b$. With uncertainty, we assume beta class priors and define the parameters $\boldsymbol{\theta}_0 = \{p_1, p_2, \ldots, p_{b-1}\}$ and $\boldsymbol{\theta}_1 = \{q_1, q_2, \ldots, q_{b-1}\}$. The bin probabilities must be valid. Thus, $\{p_1, p_2, \ldots, p_{b-1}\} \in \boldsymbol{\Theta}_0$ if and only if $0 \leq p_i \leq 1$ for $i = 1, \ldots, b-1$ and $\sum_{i=1}^{b-1} p_i \leq 1$, in which

case, $p_b = 1 - \sum_{i=1}^{b-1} p_i$. We use the Dirichlet priors

$$\pi(\boldsymbol{\theta}_0) \propto \prod_{i=1}^{b} p_i^{\alpha_i^0 - 1} \text{ and } \pi(\theta_1) \propto \prod_{i=1}^{b} q_i^{\alpha_i^1 - 1}, \tag{2.5}$$

where $\alpha_i^y > 0$. These are conjugate priors, leading to the posteriors of the same form. The effective class-conditional densities are

$$f_{\boldsymbol{\Theta}}(j|y) = \frac{U_j^y + \alpha_j^y}{n_y + \sum_{i=1}^{b} \alpha_i^y}, \tag{2.6}$$

for $y = 0, 1$, and the OBC is given by

$$\psi_{\text{OBC}}(j) = \begin{cases} 0, & \text{if } E_{\pi^*}[c] f_{\boldsymbol{\Theta}}(j|0) \geq (1 - E_{\pi^*}[c]) f_{\boldsymbol{\Theta}}(j|1); \\ 1, & \text{otherwise.} \end{cases} \tag{2.7}$$

where U_j^y denotes the observed count for class y in bin j [43]. Hereafter, $\sum_{i=1}^{b} \alpha_i^y$ is represented by α_0^y, i.e. $\alpha_0^y = \sum_{i=1}^{b} \alpha_i^y$, and is called the precision factor. In the sequel, the sub(super)-script relating to dependency on class y may be dropped; nonetheless, availability of prior knowledge for both classes is assumed.

2.2.3 Multinomial Mixture Model

In practice, data may not be labeled, due to potential tumor-tissue sample or stage heterogeneity, but still we want to classify a new sample point. A mixture model is a natural model for this scenario, assuming each sample point $\mathbf{x}_i$ arises from a mixture of multinomial distributions:

$$P_{\boldsymbol{\theta}}(\mathbf{x}_i) = \sum_{j=0}^{M-1} c_j P_{\boldsymbol{\theta}_j}(\mathbf{x}_i), \tag{2.8}$$

where M is the number of components. When there exists two components, similar to binary classification, $M = 2$. The conjugate prior distribution family for component probabilities (if unknown) is the Dirichlet distribution. In the mixture model, no closed-form analytical posterior distribution for the parameters exists, but Markov chain Monte Carlo (MCMC) methods [46] can

be employed to numerically calculate the posterior distributions. Since the conditional distributions can be calculated analytically in the multinomial mixture model, Gibbs sampling [47, 48] can be employed for the Bayesian inference. If the prior probability distribution over the component probability vector ($\boldsymbol{c} = [c_0, c_1, ..., c_M]$) is a Dirichlet distribution $\mathcal{D}(\boldsymbol{\phi})$ with parameter vector $\boldsymbol{\phi}$, the component-conditional probabilities are $\boldsymbol{\theta}_j = [p_1^j, p_2^j, \ldots, p_b^j]$, and the prior probability distribution over them is Dirichlet $\mathcal{D}(\boldsymbol{\alpha^j})$ with parameter vector $\boldsymbol{\alpha^j}$ (as in the classification problem), for $j = 1, ..., M$, the Gibbs updates are

$$y_i^{(t)} \sim P(y_i = j|\boldsymbol{c}^{(t-1)}, \boldsymbol{\theta}^{(t-1)}, \mathbf{x}_i) \propto c_j^{(t-1)} p_{\mathbf{x}_i}^{j,(t-1)}$$

$$\boldsymbol{c}^{(t)} \sim P(\boldsymbol{c}|\boldsymbol{\phi}, \boldsymbol{y}^{(t)}) = \mathcal{D}\Big(\boldsymbol{\phi} + \textstyle\sum_{i=1}^n [I_{y_i^{(t)}=1}, ..., I_{y_i^{(t)}=M}]\Big)$$

$$\boldsymbol{\theta_j}^{(t)} \sim P(\boldsymbol{\theta}_j|\mathbf{x}, \boldsymbol{y}^{(t)}, \boldsymbol{\alpha}_j) = \mathcal{D}\Big(\boldsymbol{\alpha}_j + \textstyle\sum_{i=1:y_i^{(t)}=j}^n [I_{\mathbf{x}_i=1}, ..., I_{\mathbf{x}_i=b}]\Big),$$

where the super-script in parentheses denotes the chain iteration number, I_w is one if w is true, and otherwise I_w is zero. In this framework, if the inference chain runs for Is iterations, then the numerical approximation of the OBC classification rule is

$$\psi_{\text{OBC}}(k) \approx \arg \max_{y \in \{1,...,M\}} \sum_{t=1}^{Is} c_y^{(t)} p_k^{y,(t)}. \tag{2.9}$$

Without loss of generality the summation above can be over the iterations of the chain considering burn-in and thinning.

2.2.4 Prior Construction: General Framework

In this section, we propose a general framework for prior construction. We begin with introducing a knowledge-driven prior probability:

Definition 1. *(Maximal Knowledge-driven Information Prior)*

If Π is a family of proper priors, then a maximal knowledge-driven information prior (MKDIP) is

a solution to the following optimization problem:

$$\arg\min_{\pi\in\Pi} E_{\pi}[C_{\boldsymbol{\theta}}(\xi, D)], \tag{2.10}$$

where $C_{\boldsymbol{\theta}}(\xi, D)$ is a cost function that depends on (1) $\boldsymbol{\theta}$: the random vector parameterizing the underlying probability distribution, (2) ξ: state of (prior) knowledge, and (3) D: partial observation (part of the sample data).

Alternatively, by parameterizing the prior probability as $\pi(\boldsymbol{\theta};\boldsymbol{\gamma})$, with $\boldsymbol{\gamma}\in\Gamma$ denoting the hyper-parameters, an MKDIP can be found by solving

$$\arg\min_{\boldsymbol{\gamma}\in\Gamma} E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}[C_{\boldsymbol{\theta}}(\xi, D, \boldsymbol{\gamma})]. \tag{2.11}$$

In contrast to non-informative priors, the MKDIP incorporates available prior knowledge and even *part* of the data to construct an informative prior.

The MKDIP definition is very general because we want a general framework for prior construction. The next definition specializes it to cost functions of a specific form in a constrained optimization.

Definition 2. *(MKDIP: Constrained Optimization with Additive Costs) As a special case in which $C_{\boldsymbol{\theta}}$ can be decomposed into additive terms, the cost function is of the form:*

$$C_{\boldsymbol{\theta}}(\xi, D, \boldsymbol{\gamma}) = (1-\beta)g^{(1)}_{\boldsymbol{\theta}}(\xi, \boldsymbol{\gamma}) + \beta g^{(2)}_{\boldsymbol{\theta}}(\xi, D),$$

where β is a non-negative regularization parameter. In this case, the MKDIP construction with additive costs and constraints involves solving the following optimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{\gamma}\in\Gamma} E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}\Big[(1-\beta)g^{(1)}_{\boldsymbol{\theta}}(\xi, \boldsymbol{\gamma}) + \beta g^{(2)}_{\boldsymbol{\theta}}(\xi, D)\Big] \\ &\textit{Subject to:} \quad E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}[g^{(3)}_{\boldsymbol{\theta},i}(\xi)] = 0;\ i\in\{1, ..., n_c\}, \end{aligned} \tag{2.12}$$

where $g^{(3)}_{\boldsymbol{\theta},i}$, $\forall i \in \{1, \ldots, n_c\}$, are constraints resulting from the state of knowledge ξ via a mapping:

$$\mathcal{T} : \xi \to E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}[g^{(3)}_{\boldsymbol{\theta},i}(\xi)], \forall i \in \{1, \ldots, n_c\}.$$

In the sequel, we will refer to $g^{(1)}(\cdot)$ and $g^{(2)}(\cdot)$ as the cost functions, and $g^{(3)}_i(\cdot)$'s as the knowledge-driven constraints. We begin with introducing information-theoretic cost functions, and then we propose a general set of mapping rules, denoted by $\mathcal{T}$ in Definition 2, to convert biological pathway knowledge into mathematical forms. We then consider special cases with information-theoretic cost functions.

2.2.5 Information-Theoretic Cost Functions

Instead of having least squares (or mean-squared error) as the standard cost functions in classical statistical inference problems, there is no universal cost function in the prior construction literature. That being said, we utilize several widely used cost functions in the field:

1. (Maximum Entropy) The principle of maximum-entropy (MaxEnt) for probability construction [41] leads to the least informative prior given the constraints in order to prevent adding spurious information. Under our general framework MaxEnt can be formulated by setting:

$$\beta = 0,\ g^{(1)}_{\boldsymbol{\theta}} = \ln \pi(\boldsymbol{\theta}; \boldsymbol{\gamma}),$$

   where $H[.]$ denotes the Shannon entropy.

2. (Maximal Data Information) The maximal data information prior (MDIP) introduced by Zellner [49] as a choice of the objective function is a criterion for the constructed probability distribution to remain maximally committed to the data [50]. To achieve MDIP, we can set our general framework with:

$$\beta = 0,\ g^{(1)}_{\boldsymbol{\theta}} = \ln \pi(\boldsymbol{\theta}; \boldsymbol{\gamma}) + H[x|\boldsymbol{\theta}] = \ln \pi(\boldsymbol{\theta}; \boldsymbol{\gamma}) - E_{x|\boldsymbol{\theta}}[\ln P(x|\boldsymbol{\theta})].$$

3. (Expected Mean Log-likelihood) The cost function introduced in [39] is the first one that utilizes part of the observed data for prior construction. In that, we have

$$\beta = 1,\ g_{\boldsymbol{\theta}}^{(2)} = -\ell(\boldsymbol{\theta}; D),$$

where $\ell(\boldsymbol{\theta}; D) = \frac{1}{n_D}\sum_{i=1}^{n_D} \log f(\boldsymbol{x}_i|\boldsymbol{\theta})$ is the mean log-likelihood function of the sample points used for prior construction (D), and n_D denotes the number of sample points in D. In [39], it is shown that this cost function is equivalent to the average Kullback-Leibler distance between the *unknown* distribution (empirically estimated by some part of the samples) and the uncertainty class of distributions.

As originally proposed, the preceding approaches did not involve expectation over the uncertainty class. They were extended to the general prior construction form in Definition (1), including the expectation, in [40] to produce the regularized maximum entropy prior (RMEP), the regularized maximal data information prior (RMDIP), and the regularized expected mean log-likelihood prior (REMLP). In all cases, optimization was subject to specialized constraints.

For MKDIP, we employ the same information-theoretic cost functions in the prior construction optimization framework. MKDIP-E, MKDIP-D, and MKDIP-R correspond to using the same cost functions as REMP, RMDIP, and REMLP, respectively, but with the new general types of constraints. To wit, we employ *functional information* from the signaling pathways and show that by adding these new constraints that can be readily derived from prior knowledge, we can improve both supervised (classification problem with labelled data) and unsupervised (mixture problem without labels) learning of Bayesian operators.

2.2.6 From Prior Knowledge to Mathematical Constraints

In this part, we present a general formulation for mapping the existing knowledge into a set of *constraints*. In most scientific problems, the prior knowledge is in the form of conditional probabilities. In the following, we consider a hypothetical gene network and show how each component in a given network can be converted into the corresponding inequalities as general

constraints in MKDIP optimization.

Before proceeding we would like to say something about contextual effects on regulation. Because a regulatory model is not independent of cellular activity outside the model, complete control relations such as $A \to B$ in the model, meaning that gene B is up-regulated if and only if gene A is up-regulated (after some time delay), do not necessarily translate into conditional probability statements of the form $P(X_B = 1|X_A = 1) = 1$, where X_A and X_B represent the binary gene values corresponding to genes A and B, respectively. Rather, what may be observed is $P(X_B = 1|X_A = 1) = 1 - \delta$, where $\delta > 0$. The pathway $A \to B$ need not imply $P(X_B = 1|X_A = 1) = 1$ because $A \to B$ is conditioned on the *context* of the cell, where by context we mean the overall state of the cell, not simply the activity being modeled. δ is called a *conditioning* parameter. In an analogous fashion, rather than $P(X_B = 1|X_A = 0) = 0$, what may be observed is $P(X_B = 1|X_A = 0) = \eta$, where $\eta > 0$, because there may be regulatory relations outside the model that up-regulate B. Such activity is referred to as cross-talk and η is called a *crosstalk* parameter. Conditioning and cross-talk effects can involve multiple genes and can be characterized analytically via context-dependent conditional probabilities [51].

Consider binary gene values $X_1, X_2, \ldots, X_m$ corresponding to genes $g_1, g_2, \ldots, g_m$. There are $m2^{m-1}$ conditional probabilities of the form

$$\begin{aligned} &P(X_i = k_i|X_1 = k_1, \ldots, X_{i-1} = k_{i-1}, X_{i+1} = k_{i+1}, \ldots, X_m = k_m) \\ &= a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m) \end{aligned} \tag{2.13}$$

to serve as constraints, the chosen constraints to be the conditional probabilities whose values are known (approximately). For instance, if g_2 and g_3 regulate g_1, with $X_1 = 1$ when $X_2 = 1$ and $X_3 = 0$, then, ignoring context effects,

$$a_1^1(1, 0, k_4, \ldots, k_m) = 1$$

for all $k_4, \ldots, k_m$. If, however, we take context conditioning into effect, then

$$a_1^1(1, 0, k_4, \ldots, k_m) = 1 - \delta_1(1, 0, k_4, \ldots, k_m),$$

where $\delta_1(1, 0, k_4, \ldots, k_m)$ is a conditioning parameter.

Moreover, ignoring context effects,

$$a_1^1(1, 1, k_4, \ldots, k_m) = a_1^1(0, 0, k_4, \ldots, k_m) = a_1^1(0, 1, k_4, \ldots, k_m) = 0$$

for all $k_4, \ldots, k_m$. If, however, we take crosstalk into effect, then

$$\begin{aligned} a_1^1(1, 1, k_4, \ldots, k_m) &= \eta_1(1, 1, k_4, \ldots, k_m) \\ a_1^1(0, 0, k_4, \ldots, k_m) &= \eta_1(0, 0, k_4, \ldots, k_m) \\ a_1^1(0, 1, k_4, \ldots, k_m) &= \eta_1(0, 1, k_4, \ldots, k_m), \end{aligned}$$

where $\eta_1(1, 1, k_4, \ldots, k_m)$, $\eta_1(0, 0, k_4, \ldots, k_m)$, and $\eta_1(0, 0, k_4, \ldots, k_m)$ are crosstalk parameters. In practice it is unlikely that we would know the conditioning and crosstalk parameters for all combinations of $k_4, \ldots, k_m$; rather, we might just know the average, in which case, $\delta_1(1, 0, k_4, \ldots, k_m)$ reduces to $\delta_1(1, 0)$, $\eta_1(1, 1, k_4, \ldots, k_m)$ reduces to $\eta_1(1, 1)$, etc.

In this paradigm, the constraints resulting from our state of knowledge are of the following form:

$$\begin{aligned} & g_{\boldsymbol{\theta},i}^{(3)}(\xi) = \\ & P(X_i = k_i | X_1 = k_1, \ldots, X_{i-1} = k_{i-1}, X_{i+1} = k_{i+1}, \\ & \quad \ldots, X_m = k_m) - a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m). \end{aligned} \tag{2.14}$$

The basic setting is very general and the conditional probabilities are what they are, whether or not they can be expressed in the regulatory form of conditioning or crosstalk parameters. The general scheme includes previous constraints and approaches proposed in [39] and [40] for the Gaussian and discrete setups. Moreover, in those we can drop the regulatory-set entropy because

it is replaced by the set of conditional probabilities based on the regulatory set, whether forward (master predicting slaves) or backwards (slaves predicting masters) [51].

In this paradigm, the optimization constraints take the form

$$\begin{aligned}
&a_i^{k_i}(k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_m) - \varepsilon_i(k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_m) \\
&\leq E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}[P(X_i = k_i | X_1 = k_1,\ldots,X_{i-1} = k_{i-1}, X_{i+1} = k_{i+1},\ldots,X_m = k_m)] \\
&\leq a_i^{k_i}(k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_m) + \varepsilon_i(k_1,\ldots,k_{i-1},k_{i+1},\ldots,k_m),
\end{aligned} \tag{2.15}$$

where the expectation is with respect to the uncertainty in the model parameters, that is, the distribution of the model parameter $\boldsymbol{\theta}$, and ε_i is a slackness variable. Not all will be used, depending on our prior knowledge. In fact, the general conditional probabilities will not likely be used because they will likely not be known when there are too many conditioning variables. For instance, we may not know the probability in equation (2.13), but may know the conditioning on part of the variables which can be extracted from some interaction network (e.g. biological pathways). A slackness variable can be considered for each constraint to make the constraint framework more flexible, thereby allowing potential error or uncertainty in prior knowledge (allowing potential inconsistencies in prior knowledge). When using slackness variables, these variables also become optimization parameters, and a linear function (summation of all slackness variables) times a regulatory coefficient is added to the cost function of the optimization in (2.12). In other words, when having slackness variables, the optimization in (2.12) can be written as

$$\begin{aligned}
&\arg\min_{\boldsymbol{\gamma}\in\Gamma,\boldsymbol{\varepsilon}\in\mathcal{E}} E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}\Big[\lambda_1[(1-\beta)g_{\boldsymbol{\theta}}^{(1)}(\xi,\boldsymbol{\gamma}) + \beta g_{\boldsymbol{\theta}}^{(2)}(\xi,D)] + \lambda_2\sum_{i=1}^{n_c}\varepsilon_i\Big] \\
&\text{Subject to: } -\varepsilon_i \leq E_{\pi(\boldsymbol{\theta};\boldsymbol{\gamma})}[g_{\boldsymbol{\theta},i}^{(3)}(\xi)] \leq \varepsilon_i;\ i\in\{1,...,n_c\},
\end{aligned} \tag{2.16}$$

where λ_1 and λ_2 are non-negative regularization parameters, and $\boldsymbol{\varepsilon}$ and $\mathcal{E}$ represent the vector of all slackness variables and the feasible region for slackness variables, respectively. For each slackness variable, a possible range can be defined (note that all slackness variables are non-negative).

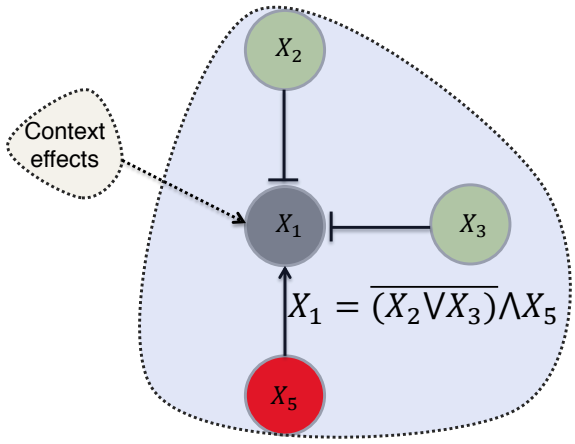


Figure 2.2: An illustrative example showing the components directly connected to gene 1. In the Boolean function $\{\text{AND, OR, NOT}\} = \{\wedge, \vee, -\}$. Based on the regulating function of gene 1, it is up-regulated if gene 5 is up-regulated and genes 2 and 3 are down-regulated.

The higher the uncertainty is about a constraint stemming from prior knowledge, the greater the possible range for the corresponding slackness variable can be (more on this in the Results and Discussion section).

The new general type of constraints discussed here introduces a formal procedure for incorporating prior knowledge. It allows the incorporation of knowledge of the functional regulations in the signaling pathways, any constraints on the conditional probabilities, and also knowledge of the cross-talk and conditioning parameters (if present), unlike the previous work in [40] where only partial information contained in the edges of the pathways is used in an ad hoc way.

2.2.7 An Illustrative Example and Connection with Conditional Entropy

Now, consider a hypothetical network depicted in Figure 2.2. For instance, suppose we know that the expression of gene g_1 is regulated by g_2, g_3, and g_5. Then we have

$$P(X_1 = 1 | X_2 = k_2, X_3 = k_3, X_5 = k_5) = a_1^1(k_2, k_3, k_5).$$

As an example,

$$P(X_1 = 1|X_2 = 1, X_3 = 1, X_5 = 0) = a_1^1(1_2, 1_3, 0_5),$$

where the notation 1_2 denotes 1 for the second gene. Further, we might not know $a_1(k_2, k_3, k_5)$ for all combinations of k_2, k_3, k_5. Then we use the ones that we know. In the case of conditioning with g_2, g_3, and g_5 regulating g_1, with g_1 on if the others are on,

$$a_1^1(1_2, 1_3, 1_5) = 1 - \delta_1(1_2, 1_3, 1_5).$$

If limiting to 3-gene predictors, g_3, and g_5 regulate g_1, with g_1 on if the other two are on, then

$$a_1^1(k_2, 1_3, 1_5) = 1 - \delta_1(k_2, 1_3, 1_5),$$

meaning that the conditioning parameter depends on whether $X_2 = 0$ or 1.

Now, considering the conditional entropy, assuming that $\delta_1 = \max_{(k_2,k_3,k_5)} \delta_1(k_2, k_3, k_5)$ and $\delta_1 < 0.5$, we may write

$$\begin{aligned}
&H[X_1|X_2, X_3, X_5] = \\
&-\Big[\sum_{\mathcal{X}_2,\mathcal{X}_3,\mathcal{X}_5} [P(X_1 = 0|X_2 = x_2, X_3 = x_3, X_5 = x_5)\times \\
&P(X_2 = x_2, X_3 = x_3, X_5 = x_5) \log[P(X_1 = 0|X_2 = x_2, X_3 = x_3, X_5 = x_5)] \\
&+ P(X_1 = 1|X_2 = x_2, X_3 = x_3, X_5 = x_5)\times \\
&P(X_2 = x_2, X_3 = x_3, X_5 = x_5) \log[P(X_1 = 1|X_2 = x_2, X_3 = x_3, X_5 = x_5)]]\Big] \\
&\qquad \leq h(\delta_1),
\end{aligned}$$

where $h(\delta) = -[\delta \log(\delta) + (1 - \delta) \log(1 - \delta)]$. Hence, bounding the conditional probabilities, the

conditional entropy is in turn bounded by $h(\delta_1)$:

$$\lim_{\delta_1 \to 0^+} H[X_1|X_2, X_3, X_5] = 0.$$

It should be noted that constraining $H[X_1|X_2, X_3, X_5]$ would not necessarily constrain the conditional probabilities, and may be considered as a more relaxed type of constraints. But, for example, in cases where there is no knowledge about the status of a gene given its regulator genes, constraining entropy is the only possible approach.

In our illustrative example, if we assume that the Boolean regulating function of X_1 is known as shown in Figure 2.2 and context effects exist, then the following knowledge constraints can be extracted from the pathway and regulating function:

$$\begin{aligned}
a_1^0(k_2, k_3, 0_5) &= 1 - \delta_1(k_2, k_3, 0_5) \\
a_1^0(k_2, 1_3, k_5) &= 1 - \delta_1(k_2, 1_3, k_5) \\
a_1^0(1_2, k_3, k_5) &= 1 - \delta_1(1_2, k_3, k_5) \\
a_1^1(0_2, 0_3, 1_5) &= 1 - \delta_1(0_2, 0_3, 1_5).
\end{aligned}$$

Now if we assume that the context does not affect the value of X_1, i.e. the value of X_1 can be fully determined by knowing the values of X_2, X_3, and X_5, then we have the following equations:

$$a_1^0(k_2, k_3, 0_5) = P(X_1 = 0|X_5 = 0) = 1 \tag{2.17a}$$

$$a_1^0(k_2, 1_3, k_5) = P(X_1 = 0|X_3 = 1) = 1 \tag{2.17b}$$

$$a_1^0(1_2, k_3, k_5) = P(X_1 = 0|X_2 = 1) = 1 \tag{2.17c}$$

$$a_1^1(0_2, 0_3, 1_5) = P(X_1 = 1|X_2 = 0, X_3 = 0, X_5 = 1) = 1. \tag{2.17d}$$

It can be seen from the equations above that for some setups of the regulator values, only a subset of them determines the value of X_1, regardless of the other regulator values. If we assume that the

value of X_5 cannot be observed, for example X_5 is an extracellular signal that cannot be measured in gene expression data and thereafter X_5 is not in the features of our data, the only constraints relevant to the feature-label distribution that can be extracted from the regulating function knowledge will be

$$a_1^0(k_2, 1_3, k_5) = P(X_1 = 0|X_3 = 0) = 1 \tag{2.18}$$
$$a_1^0(1_2, k_3, k_5) = P(X_1 = 0|X_2 = 0) = 1.$$

2.2.8 Special Case of Dirichlet Distribution

Fixing the value of a single gene, being ON or OFF (i.e. $X_i = 0$ or $X_i = 1$, respectively), corresponds to a partition of the states, $\mathcal{X} = \{1, \ldots, b\}$. Here, the portions of $\mathcal{X}$ for which $(X_i = k_1, X_j = k_2)$ and $(X_i \neq k_1, X_j = k_2)$, for any $k_1, k_2 \in \{0, 1\}$, are denoted by $\mathcal{X}^{i,j}(k_1, k_2)$ and $\mathcal{X}^{i,j}(k_1^c, k_2)$, respectively. For the Dirichlet distribution, where $\boldsymbol{\theta} = \boldsymbol{p}$ and $\boldsymbol{\gamma} = \boldsymbol{\alpha}$, the constraints on the expectation over the conditional probability in (2.15) can be explicitly written as functions of the prior probability parameters (hyperparameters). For the parameter of the Dirichlet distribution, a vector $\boldsymbol{\alpha}$ indexed by $\mathcal{X}$, we denote the variable indicating the summation of its entities in $\mathcal{X}^{i,j}(k_1, k_2)$ by $\overline{\alpha}^{i,j}(k_1, k_2) = \sum_{k \in \mathcal{X}^{i,j}(k_1,k_2)} \alpha_k$. The notation can be easily extended for the cases having more than two fixed genes. In this setup, if the set of random variables corresponding to genes other than g_i and the vector of their corresponding values are shown by $\tilde{X}_i$ and $\tilde{x}_i$, respectively, the expectation over the conditional probability in (2.15) is [40]:

$$\begin{aligned}
&E_{\boldsymbol{p}}[P(X_i = k_i|X_1 = k_1, \ldots, X_{i-1} = k_{i-1}, X_{i+1} = k_{i+1}, \ldots, X_m = k_m)] = \\
&E_{\boldsymbol{p}}\Big[\frac{\sum_{k \in \mathcal{X}^{i,\tilde{X}_i}(k_i,\tilde{x}_i)} p_k}{\sum_{k \in \mathcal{X}^{i,\tilde{X}_i}(k_i,\tilde{x}_i)} p_k + \sum_{k \in \mathcal{X}^{i,\tilde{X}_i}(k_i^c,\tilde{x}_i)} p_k}\Big] = \\
&\frac{\overline{\alpha}^{i,\tilde{X}_i}(k_i, \tilde{x}_i)}{\overline{\alpha}^{i,\tilde{X}_i}(k_i, \tilde{x}_i) + \overline{\alpha}^{i,\tilde{X}_i}(k_i^c, \tilde{x}_i)},
\end{aligned} \tag{2.19}$$

where the summation in the numerator and the first summation in the denominator of the second equality is over the states (bins) for which $(X_i = k_i, \tilde{X}_i = \tilde{x}_i)$, and the second summation in the

denominator is over the states (bins) for which $(X_i = k_i^c, \tilde{X}_i = \tilde{x}_i)$.

If there exists a set of genes that completely determines the value of gene g_i (or only a specific setup of their values that determines the value, as we had in our illustrative example in equations (2.17)), then the constraints on the conditional probability conditioned on all the genes other than g_i can be changed to be conditioned on that set only. Specifically, let $\boldsymbol{R}_i$ denote the set of random variables corresponding to such a set of genes/proteins and suppose there exists a specific setup of their values $\boldsymbol{r}_i$ that completely determines the value of gene g_i. If the set of all random variables corresponding to the genes/proteins other than X_i and $\boldsymbol{R}_i$ is denoted by $\boldsymbol{B}_i = \tilde{X}_{(i,\boldsymbol{R}_i)}$, then the constraints on the conditional probability can be written as

$$E_{\boldsymbol{p}}[P(X_i = k_i|\boldsymbol{R}_i = \boldsymbol{r}_i)] = E_{\boldsymbol{p}}\Big[\frac{\sum_{k\in\mathcal{X}^{i,\boldsymbol{R}_i}(k_i,\boldsymbol{r}_i)} p_k}{\sum_{k\in\mathcal{X}^{i,\boldsymbol{R}_i}(k_i,\boldsymbol{r}_i)} p_k + \sum_{k\in\mathcal{X}^{i,\boldsymbol{R}_i}(k_i^c,\boldsymbol{r}_i)} p_k}\Big] = \frac{\overline{\alpha}^{i,\boldsymbol{R}_i}(k_i,\boldsymbol{r}_i)}{\overline{\alpha}^{i,\boldsymbol{R}_i}(k_i,\boldsymbol{r}_i) + \overline{\alpha}^{i,\boldsymbol{R}_i}(k_i^c,\boldsymbol{r}_i)}, \tag{2.20}$$

where $\mathcal{X}^{i,\boldsymbol{R}_i}(k_i,\boldsymbol{r}_i)$ is the partition containing all the states corresponding to $X_i = k_i$, $\boldsymbol{R}_i$ fixed at vector of values $\boldsymbol{r}_i$, and all possible vectors of values of $\boldsymbol{B}_i$.

For a multinomial model with a Dirichlet prior distribution, a constraint on the conditional probabilities translates into a constraint on the above expectation over the conditional probabilities (as in (2.15)). In our illustrative example and from the equations in (2.17), there are four of these constraints on the conditional probability for gene g_1. For example, in the second constraint from the second line of equation (2.17) (equation (2.17b)), $X_i = X_1$, $k_i = 0$, $\boldsymbol{R}_i = \{X_3\}$, $\boldsymbol{r}_i = [0]$, and $\boldsymbol{B}_i = \{X_2, X_5\}$. One might have several constraints for each gene extracted from its regulatory function (more on extracting general constraints from regulating functions in the Results and Discussion section).

2.3 Results and Discussion

The performance of the proposed general prior construction framework with different types of objective functions and constraints is examined and compared with other methods on two pathways, a mammalian cell-cycle pathway and a pathway involving the gene TP53. Here we employ Boolean network modeling of genes/proteins (hereafter referred to as entities or nodes) [52] with perturbation (BNp). A Boolean Network with p nodes (genes/proteins) is defined as $B = (V, F)$, where V represents the set of entities (genes/proteins) $\{v_1, \ldots, v_p\}$, and F is the set of Boolean predictor functions $\{f_1, \ldots, f_p\}$. At each step in a BNp, a decision is made by a Bernoulli random variable with the success probability equal to the perturbation probability, p_{pert}, as to whether a node value is determined by perturbation of randomly flipping its value or by the logic model imposed from the interactions in the signaling pathways. A BNp with a positive perturbation probability can be modeled by an ergodic Markov chain, and possesses a steady-state distribution (SSD) [53]. The performance of different prior construction methods can be compared based on the expected true error of the optimal Bayesian classifiers designed with those priors, and also by comparing these errors with some other well known classification methods. Another comparison metric of prior construction methods is the expected norm of the difference between the true parameters and the posterior mean of these parameters inferred using the constructed prior distributions. Here, the true parameters are the vectors of the true class-conditional SSDs, i.e. the vectors of the true class-conditional bin probabilities of the BNp.

Moreover, the performance of the proposed framework is compared with other methods on a publicly available gene expression dataset of non-small cell lung cancer when combined with the existing prior knowledge on relevant signaling pathways.

2.3.1 Mammalian Cell Cycle Classification

A Boolean logic regulatory network for the dynamical behavior of the cell cycle of normal mammalian cells is proposed in [2]. Figure 2.3(a) shows the corresponding pathways. In normal cells, cell division is coordinated via extracellular signals controlling the activation of CycD. Rb is

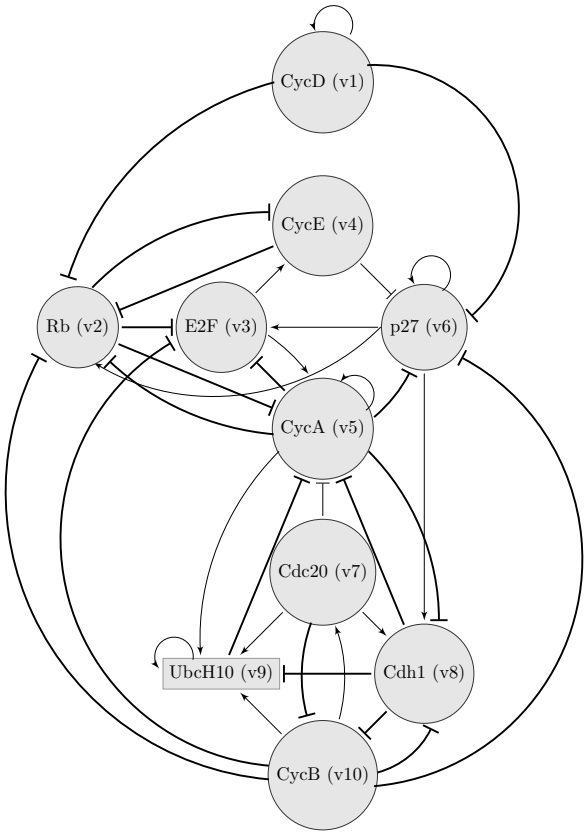
a tumor suppressor gene and is expressed when the inhibitor cyclins are not present. Expression of p27 blocks the action of CycE or CycA, and lets the tumor-suppressor gene Rb be expressed even in the presence of CycE and CycA, and results in a stop in the cell cycle. Therefore, in the wild-type cell-cycle network, expressing p27 lets the cell cycle stop. But following the proposed mutation in [2], for the mutated case, p27 is always inactive (i.e. can never be activated), thereby creating a situation where both CycD and Rb might be inactive and the cell can cycle in the absence of any growth factor.

The full functional regulations in the cell-cycle Boolean network are shown in Table 2.1. Fol-

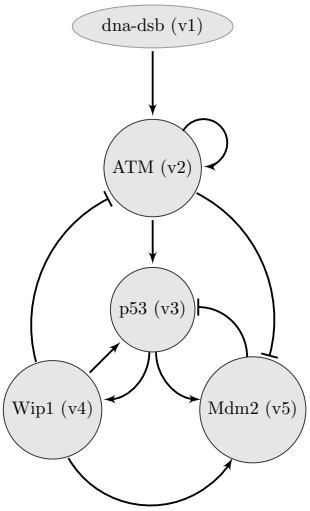
Table 2.1: Boolean regulating functions of normal mammalian cell cycle adapted from [2]. In the Boolean functions $\{$AND, OR, NOT$\} = \{\wedge, \vee, -\}$.

Gene	Node name	Boolean regulating function
CycD	v_1	Extracellular signal
Rb	v_2	$(\overline{v_1} \wedge \overline{v_4} \wedge \overline{v_5} \wedge \overline{v_{10}}) \vee (v_6 \wedge \overline{v_1} \wedge \overline{v_{10}})$
E2F	v_3	$(\overline{v_2} \wedge \overline{v_5} \wedge \overline{v_{10}}) \vee (v_6 \wedge \overline{v_2} \wedge \overline{v_{10}})$
CycE	v_4	$(v_3 \wedge \overline{v_2})$
CycA	v_5	$(v_3 \wedge \overline{v_2} \wedge \overline{v_7} \wedge \overline{(v_8 \wedge v_9)}) \vee (v_5 \wedge \overline{v_2} \wedge \overline{v_7} \wedge \overline{(v_8 \wedge v_9)})$
p27	v_6	$(\overline{v_1} \wedge \overline{v_4} \wedge \overline{v_5} \wedge \overline{v_{10}}) \vee (v_6 \wedge \overline{(v_4 \wedge v_5)} \wedge \overline{v_{10}} \wedge \overline{v_1})$
Cdc20	v_7	v_{10}
Cdh1	v_8	$(\overline{v_5} \wedge \overline{v_{10}}) \vee (v_7) \vee (v_6 \wedge \overline{v_{10}})$
UbcH10	v_9	$(\overline{v_8}) \vee (v_8 \wedge v_9 \wedge (v_7 \vee v_5 \vee v_{10}))$
CycB	v_{10}	$(\overline{v_7} \wedge \overline{v_8})$

lowing [40], for the binary classification problem, $y = 0$ corresponds to the normal system functioning based on Table 2.1, and $y = 1$ corresponds to the mutated (cancerous) system where CycD, p27, and Rb are permanently down-regulated (are stuck at zero), which creates a situation where the cell cycles even in the absence of any growth factor. The perturbation probability is set to 0.01 and 0.05 for the normal and mutated system, respectively. A BNp has a transition probability matrix (TPM), and as mentioned earlier, with positive perturbation probability can be modeled by an ergodic Markov chain, and possesses a SSD [53]. Here, each class



(a) Mammalian cell-cycle pathway



(b) TP53 pathway

Figure 2.3: Signaling pathways corresponding to Tables 2.1 and 2.2. Signaling pathways for: 2.3(a) the normal mammalian cell cycle (corresponding to Table 2.1) and 2.3(b) a simplified pathway involving TP53 (corresponding to Table 2.2)

has a vector of steady-state bin probabilities, resulting from the regulating functions of its corresponding BNp and the perturbation probability. The constructed SSDs are further marginalized to a subset of seven genes to prevent trivial classification scenarios. The final feature vector is $\mathbf{x} = [\mathrm{E2F}, \mathrm{CycE}, \mathrm{CycA}, \mathrm{Cdc20}, \mathrm{Cdh1}, \mathrm{UbcH10}, \mathrm{CycB}]$, and the state space size is $2^7 = 128$. The true parameters for each class are the final class-conditional steady-state bin probabilities, $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ for the normal and mutated systems, respectively, which are utilized for taking samples.

2.3.2 Classification Problem corresponding to TP53

TP53 is a tumor suppressor gene involved in various biological pathways [40]. Mutated p53 has been observed in almost half of the common human cancers [54], and in more than 90% of patients with severe ovarian cancer [55]. A simplified pathway involving TP53, based on logic in [3], is shown in Figure 2.3(b). DNA double-strand break affects the operation of these pathways, and the Boolean network modeling of these pathways under this uncertainty has been studied in [3, 55]. The full functional regulations are shown in Table 2.2. Following [40], two scenarios, dna-dsb=0

Table 2.2: Boolean regulating functions corresponding to the pathway in Figure 2.3(b) adapted from [3]. In the Boolean functions {AND, OR, NOT} $= \{\wedge, \vee, -\}$.

Gene	Node name	Boolean regulating function
$\mathrm{dna-dsb}$	v_1	Extracellular signal
ATM	v_2	$\overline{v_4} \wedge (v_2 \vee v_1)$
$\mathrm{P53}$	v_3	$\overline{v_5} \wedge (v_2 \vee v_4)$
$\mathrm{Wip1}$	v_4	v_3
$\mathrm{Mdm2}$	v_5	$\overline{v_2} \wedge (v_3 \vee v_4)$

and dna-dsb=1, weighted by 0.95 and 0.05, are considered and the SSD of the normal system is constructed based on the ergodic Markov chain model of the BNp with the regulating functions in Table 2.2 by assuming the perturbation probability 0.01. The SSD for the mutated (cancerous) case is constructed by assuming a permanent down regulation of TP53 in the BNp, and perturbation

probability 0.05. Knowing that dna-dsb is not measurable, and to avoid trivial classification situations, the SSDs are marginalized to a subset of three entities $\mathbf{x} = [\mathrm{ATM}, \mathrm{Wip1}, \mathrm{Mdm2}]$. The state space size in this case is $2^3 = 8$. The true parameters for each class are the final class-conditional steady-state bin probabilities, $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ for the normal and mutated systems, respectively, which are used for data generation.

2.3.3 Extracting General Constraints from Regulating Functions

If knowledge of the regulating functions exists, it can be used in the general constraint framework of the MKDIP, i.e. it can be used to constrain the conditional probabilities. In other words, the knowledge about the regulating function of gene i can be used to set $\varepsilon_i(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$, and $a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$ in the general form of constraints in (2.15). If the true regulating function of gene i is known, and it is not context sensitive, then the conditional probability of its status, $a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$, is known for sure, and $\delta_i(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m) = 0$. But in reality, the true regulating functions are not known, and are also context sensitive. The dependence on the context translates into $\delta_i(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$ being greater than zero. The greater the context effect on the gene status, the larger δ_i is. Moreover, the uncertainty over the regulating function is captured by the slackness variables $\varepsilon_i(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$ in (2.15). In other words, the uncertainty is translated to the possible range of the slackness variable values in the prior construction optimization framework. The higher the uncertainty is, the greater the range should be in the optimization framework. In fact, slackness variables make the whole constraint framework consistent, even for cases where the conditional probability constraints imposed by prior knowledge are not completely in line with each other, and guarantee the existence of a solution.

As an example, for the classification problems of the mammalian cell-cycle network and the TP53 network, assuming the regulating functions in Tables 2.1 and 2.2 are the true regulating functions, the context effect can be observed in the dependence of the output of the Boolean regulating functions in the tables on the extracellular signals, non-measurable entities, and the genes that have been marginalized out in our setup. In the absence of quantitative knowledge about the

context effect, i.e. $a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$ for all possible setups of the regulator values, one can impose only those with such knowledge. For example, in the mammalian cell-cycle network, CycB's regulating function only depends on the values included in the observed feature set; therefore the conditional probabilities are known for all regulator value setups. But for CycE the regulating function depends on Rb, which is marginalized out in our feature set, and also itself depends on an extracellular signal. Hence, the conditional probability constraints for CycE are known only for the setup of the features that determine the output of the Boolean regulating function independent of the other regulator values.

In our comparison analysis, $a_i^{k_i}(k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_m)$ for each gene/protein in (2.15) is set to one for the feature value setups that determine the Boolean regulating output regardless of the context. But since the observed data are not fully described by these functions, and the system has uncertainty, we let the possible range for the slackness variables in (2.15) be $[0, 1)$.

We now continue the examples on two of the mammalian cell-cycle network nodes, CycB and CycE. For CycB the following constraints on the prior distribution are extracted from its regulating function:

$$E_{\boldsymbol{p}}[P(v_{10} = 0|v_8 = 1)] \geq 1 - \epsilon_1$$

$$E_{\boldsymbol{p}}[P(v_{10} = 0|v_7 = 1)] \geq 1 - \epsilon_2$$

$$E_{\boldsymbol{p}}[P(v_{10} = 1|v_7 = 0, v_8 = 0)] \geq 1 - \epsilon_3.$$

For CycE, one of its regulators is Rb (v_2), which is not included in the feature set, i.e. not observed, but is known to be down-regulated in the mutated (cancerous) case. Thus, the set of constraints extracted from the regulating function of CycE for the normal case includes only

$$E_{\boldsymbol{p}}[P(v_4 = 0|v_3 = 0)] \geq 1 - \epsilon_1$$

and for the mutated case consists of

$$E_{\boldsymbol{p}}[P(v_4 = 0|v_3 = 0)] \geq 1 - \epsilon_1$$

$$E_{\boldsymbol{p}}[P(v_4 = 1|v_3 = 1)] \geq 1 - \epsilon_2.$$

As another example, for the TP53 network, the set of constraints extracted from the regulating functions in Table 2.2 for the normal case are shown in the left panel of Table 2.3. The first and

Table 2.3: The set of constraints extracted from the regulating functions and pathways for the TP53 network. Constraints extracted from the Boolean regulating functions in Table 2.2 corresponding to the pathway in Figure 2.3(b) used in MKDIP-E, MKDIP-D, MKDIP-R (left). Constraints extracted from the pathway in Figure 2.3(b) used in RMEP, RMDIP, REMLP (right).

(a) MKDIP Constraints

Node	Constraint
v_2	$E_{\boldsymbol{p}}[P(v_2 = 0 \mid v_4 = 1)] \geq 1 - \epsilon_1$
v_2	$E_{\boldsymbol{p}}[P(v_2 = 1 \mid v_4 = 0)] \geq 1 - \epsilon_2$
v_5	$E_{\boldsymbol{p}}[P(v_5 = 0 \mid v_2 = 1)] \geq 1 - \epsilon_3$
v_5	$E_{\boldsymbol{p}}[P(v_5 = 1 \mid v_2 = 0, v_4 = 1)] \geq 1 - \epsilon_4$

(b) Constraints in Methods of [40]

Node	Constraint
v_2	$E_{\boldsymbol{p}}[P(v_2 = 0 \mid v_4 = 1)] \geq 1 - \epsilon_1$
v_5	$E_{\boldsymbol{p}}[P(v_5 = 1 \mid v_2 = 0, v_4 = 1)] \geq 1 - \epsilon_2$

second constraints for MKDIP in the left panel of Table 2.3 come from the regulating function of v_2 in Table 2.2. Although v_1 is an extracellular signal, the value of v_4 imposes two constraints on the value of v_2. But the regulating function of v_4 in Table 2.2 only depends on v_3, which is not included in our feature set, so we have no imposed constraints on the conditional probability from its regulating function. The other two constraints for MKDIP in the left panel of Table 2.3 are extracted from the regulating function of v_5 in Table 2.2. Although v_3 is not included in the observed features, for two setups of its regulators, ($v_2 = 1$) and ($v_2 = 0, v_4 = 1$), the value of v_5 can be determined, so the constraint is imposed on the prior distribution from the regulating function. For comparison, the constraints extracted from the pathway in Figure 2.3(b) based on the method of [40] are provided in the right panel of Table 2.3.

2.3.4 Performance Comparison in Classification Setup

For both the mammalian cell cycle and TP53 problems, the performance of 11 methods are compared for classification performance. OBC with the Jeffreys' prior, OBC with our previous prior construction methods in [40] (RMEP, RMDIP, REMLP), OBC with our proposed general

framework of constraints (MKDIP-E, MKDIP-D, MKDIP-R), and also well known methods including Histogram rule (Hist), CART[56], Random Forest (RF)[57], and Support Vector Machine classification (SVM) [58, 59]. Also, for all the Bayesian methods using OBC, the posterior mean of the parameters' distance from the true parameters is calculated and compared. The samples from the true distributions are stratified fixing two different class prior probabilities. Following [40], we assume that $\max_i p_i^{y,true}$, for $y \in \{0,1\}$, is known within a $+/-5\%$ interval (can come from existing population statistics in practice). Two simulation scenarios are performed: one assuming the complete knowledge of the optimal precision factors [40] $\alpha_0^y = \sum_{i=1}^b \alpha_i^y, y \in \{0,1\}$ for prior construction methods (oracle precision factor); and the other estimating the optimal precision factor from the observed data itself. Two class prior probabilities, $c = 0.6$ and $c = 0.5$, are considered. Along with the true class-conditional SSDs of the two classes, the corresponding Bayes error corresponds to the best performance that any classification rule for that classification problem (feature-label distribution) can yield. Fixing c and the true class-conditional bin probabilities, n sample points by stratified sampling ($n_0 = \lceil cn \rceil$ sample points from class 0 and $n_1 = n - n_0$ sample points from class 1) are taken for prior construction (if used by the method), classifier training, and posterior distribution calculations. Then the designed classifier's true classification error is calculated for all classification methods. The posterior mean of parameter distance from the true parameter (true steady-state bin probabilities vector) is calculated based on $\sum_{y=0}^1 ||\boldsymbol{\alpha}^{y*}/\alpha_0^{y*} - \boldsymbol{p}^y||^2$, where $\boldsymbol{\alpha}^{y*}$ and $\boldsymbol{p}^y$ represent the parameters of the posterior distribution and true bin probabilities vector for class y, respectively. For each fixed c and n, 800 Monte Carlo repetitions are done to calculate the expected classification errors and posterior distances from the true parameters for each parameter setup. For REMLP and MKDIP-R, which use a fraction of data in their prior construction procedure, 10 data points from each class are used for prior construction, and all for the inference and posterior calculation (here the number of data points used for prior construction is not fine-tuned, but a small number is chosen to avoid overfitting). The overall procedure taken for a fixed classification problem and a fixed sample size (fixed n) in each Monte Carlo repetition is as follows:

- The true bin probabilities $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ are fixed.
- n_0 and n_1 are determined using c as $n_0 = \lceil cn \rceil$ and $n - n_0$.
- Observations (training data) are randomly sampled from the multinomial distribution for each class, i.e. $(U_1^y, ..., U_b^y) \sim \mathcal{M}ult(\boldsymbol{p}^y; n_y)$, for $y \in \{0, 1\}$.
- 10 data points are randomly taken from the training data points of each class to be used in the prior construction methods that utilize partial data (REMLP and MKDIP-R)
- All the classification rules are trained based on their constructed prior (if applicable to that classification rule) and the training data.
- The classification errors associated with the classifiers are computed using $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$. Also for the Bayesian methods, the posterior probability mass (mean) distance from the true parameters (true bin probabilities, $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$) is calculated.

The regularization parameter λ_1 is set to 0.5, and λ_2 is set to 0.25 and 0.5 for the mammalian cell cycle classification problem and the TP53 classification problem, respectively. The results of expected classification error and posterior mean distance from the true parameters for the mammalian cell-cycle network are shown in Tables 2.4 and 2.6, respectively. Tables 2.5 and 2.7 contain the results of expected classification error and posterior mean distance from the true parameters for the TP53 network.

The best performance (with the lowest error in Tables 2.4 and 2.5, and lowest distance in Tables 2.6 and 2.7) for each sample size, are written in bold. For the mammalian cell-cycle network, MKDIP methods show the best (or as good as the best) performance in all the scenarios in terms of both the expected classification error and posterior parameter estimates. For the TP53 network, MKDIP methods show the best performances in posterior parameter estimates, and are competitive with the previous knowledge-driven prior construction methods in terms of the expected classification error.

Table 2.4: Expected true error of different classification rules for the mammalian cell-cycle network. The constructed priors are considered using two precision factors: optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$, where the minimum achievable error (Bayes error) is denoted by Err_{Bayes}. The lowest error for each sample size is written in bold.

(a) $c = 0.5$, optimal precision factor,$Err_{Bayes} = 0.2648$

Method/ n	30	60	90	120	150
Hist	0.3710	0.3423	0.3255	0.3155	0.3081
CART	0.3326	0.3195	0.3057	0.3031	0.2975
RF	0.3359	0.3160	0.3015	0.2991	0.2933
SVM	0.3359	0.3112	**0.2977**	0.2959	0.2940
Jeffreys'	0.3710	0.3423	0.3255	0.3155	0.3081
RMEP	0.3236	0.3070	0.3010	0.2946	0.2910
RMDIP	0.3236	0.3070	0.3010	0.2946	0.2910
REMLP	0.3425	0.3264	0.3146	0.3067	0.3011
MKDIP-E	0.3221	0.3070	0.3010	0.2949	0.2910
MKDIP-D	0.3232	0.3070	0.3010	0.2952	0.2910
MKDIP-R	**0.3149**	**0.3028**	0.2985	**0.2943**	**0.2907**

(b) $c = 0.5$, estimated precision factor, $Err_{Bayes} = 0.2648$

Method/ n	30	60	90	120	150
Hist	0.3710	0.3423	0.3255	0.3155	0.3081
CART	0.3326	0.3195	0.3057	0.3031	0.2975
RF	0.3359	0.3160	0.3015	0.2991	0.2933
SVM	0.3359	0.3112	0.2977	0.2959	0.2940
Jeffreys'	0.3710	0.3423	0.3255	0.3155	0.3081
RMEP	0.3315	0.3059	0.2985	0.2963	0.2930
RMDIP	0.3314	0.3060	0.2986	0.2965	0.2931
REMLP	0.3488	0.3352	0.3202	0.3101	0.3048
MKDIP-E	0.3313	0.3056	0.2982	0.2962	0.2929
MKDIP-D	0.3315	0.3061	0.2986	0.2965	0.2931
MKDIP-R	**0.3205**	**0.3041**	**0.2969**	**0.2947**	**0.2919**

(c) $c = 0.6$, optimal precision factor, $Err_{Bayes} = 0.31$

Method/ n	30	60	90	120	150
Hist	0.3622	0.3608	0.3624	0.3641	0.3652
CART	0.3554	0.3556	0.3507	0.3510	0.3447
RF	0.3524	0.3514	0.3467	0.3476	0.3420
SVM	0.3735	0.3684	0.3615	0.3602	0.3544
Jeffreys'	0.3620	0.3559	0.3519	0.3502	0.3472
RMEP	**0.3415**	0.3385	**0.3394**	**0.3390**	**0.3386**
RMDIP	**0.3415**	**0.3383**	**0.3394**	**0.3390**	**0.3386**
REMLP	0.3666	0.3625	0.3587	0.3558	0.3530
MKDIP-E	**0.3415**	0.3384	**0.3394**	**0.3390**	**0.3386**
MKDIP-D	**0.3415**	0.3386	**0.3394**	**0.3390**	**0.3386**
MKDIP-R	0.3437	0.3409	0.3404	0.3401	0.3389

(d) $c = 0.6$, estimated precision factor, $Err_{Bayes} = 0.31$

Method/ n	30	60	90	120	150
Hist	0.3622	0.3608	0.3624	0.3641	0.3652
CART	0.3554	0.3556	0.3507	0.3510	0.3447
RF	0.3524	0.3514	0.3467	0.3476	0.3420
SVM	0.3735	0.3684	0.3615	0.3602	0.3544
Jeffreys'	0.3620	0.3559	0.3519	0.3502	0.3472
RMEP	0.3528	0.3415	0.3407	0.3388	0.3378
RMDIP	0.3529	0.3415	0.3408	0.3388	0.3378
REMLP	0.3700	0.3650	0.3603	0.3578	0.3546
MKDIP-E	0.3525	**0.3413**	**0.3405**	**0.3387**	**0.3377**
MKDIP-D	0.3532	0.3418	0.3409	0.3389	0.3379
MKDIP-R	**0.3486**	0.3416	0.3416	0.3402	0.3387

Table 2.5: Expected true error of different classification rules for the TP53 network. The constructed priors are considered using two precision factors: optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$, where the minimum achievable error (Bayes error) is denoted by Err_{Bayes}. The lowest error for each sample size is written in bold.

(a) $c = 0.5$, optimal precision factor,$Err_{Bayes} = 0.3146$

Method/ n	15	30	45	60	75
Hist	0.3586	0.3439	0.3337	0.3321	0.3296
CART	0.3633	0.3492	0.3350	0.3314	0.3295
RF	0.3791	0.3574	0.3461	0.3400	0.3362
SVM	0.3902	0.3481	0.3433	0.3324	0.3322
Jeffreys'	0.3809	0.3439	0.3457	0.3321	0.3334
RMEP	0.3399	0.3392	0.3360	0.3315	0.3328
RMDIP	0.3399	0.3392	0.3360	0.3315	0.3328
REMLP	0.3405	**0.3340**	**0.3320**	**0.3292**	0.3287
MKDIP-E	**0.3397**	0.3398	0.3351	0.3306	0.3297
MKDIP-D	**0.3397**	0.3398	0.3347	0.3306	0.3297
MKDIP-R	0.3435	0.3354	0.3321	0.3295	**0.3283**

(b) $c = 0.5$, estimated precision factor, $Err_{Bayes} = 0.3146$

Method/ n	15	30	45	60	75
Hist	0.3586	0.3439	**0.3337**	0.3321	0.3296
CART	0.3633	0.3492	0.3350	**0.3314**	0.3295
RF	0.3791	0.3574	0.3461	0.3400	0.3362
SVM	0.3902	0.3481	0.3433	0.3324	0.3322
Jeffreys'	0.3809	0.3439	0.3457	0.3321	0.3334
RMEP	0.3791	0.3489	0.3377	0.3329	0.3302
RMDIP	0.3789	0.3490	0.3378	0.3329	0.3302
REMLP	**0.3417**	**0.3372**	0.3350	0.3318	0.3292
MKDIP-E	0.3675	0.3470	0.3373	0.3326	0.3298
MKDIP-D	0.3668	0.3472	0.3374	0.3327	0.3298
MKDIP-R	0.3471	0.3402	0.3349	0.3316	**0.3287**

(c) $c = 0.6$, optimal precision factor, $Err_{Bayes} = 0.2691$

Method/ n	15	30	45	60	75
Hist	0.3081	0.2965	0.2906	0.2883	0.2846
CART	0.3173	0.2988	0.2882	0.2846	**0.2796**
RF	0.3333	0.3035	0.2946	0.2850	0.2842
SVM	0.3322	0.3091	0.2991	0.2926	0.2857
Jeffreys'	0.3105	0.2936	0.2860	**0.2828**	0.2819
RMEP	**0.2924**	0.2922	0.2847	0.2843	0.2835
RMDIP	**0.2924**	0.2922	0.2847	0.2843	0.2835
REMLP	0.3003	**0.2908**	0.2869	0.2839	0.2832
MKDIP-E	**0.2924**	0.2909	**0.2837**	0.2851	0.2837
MKDIP-D	**0.2924**	0.2909	**0.2837**	0.2851	0.2837
MKDIP-R	0.3032	0.2917	0.2868	0.2843	0.2825

(d) $c = 0.6$, estimated precision factor, $Err_{Bayes} = 0.2691$

Method/ n	15	30	45	60	75
Hist	0.3081	0.2965	0.2906	0.2883	0.2846
CART	0.3173	0.2988	0.2882	0.2846	**0.2796**
RF	0.3333	0.3035	0.2946	0.2850	0.2842
SVM	0.3322	0.3091	0.2991	0.2926	0.2857
Jeffreys'	0.3105	0.2936	**0.2860**	**0.2828**	0.2819
RMEP	0.3346	0.3024	0.2894	0.2860	0.2823
RMDIP	0.3344	0.3023	0.2895	0.2858	0.2823
REMLP	**0.3054**	**0.2930**	0.2910	0.2870	0.2850
MKDIP-E	0.3341	0.3025	0.2898	0.2864	0.2822
MKDIP-D	0.3347	0.3024	0.2898	0.2862	0.2822
MKDIP-R	0.3096	0.2981	0.2910	0.2869	0.2849

Table 2.6: Expected difference between the true model (for mammalian cell-cycle network) and estimated posterior probability masses. Optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$. The lowest distance for each sample size is written in bold

(a) $c = 0.5$, optimal precision factor

Method/ n	30	60	90	120	150
Jeffreys'	0.2155	0.1578	0.1300	0.1134	0.1010
RMEP	0.1591	0.1293	0.1126	0.1020	0.0912
RMDIP	0.1591	0.1294	0.1126	0.1020	0.0912
REMLP	0.1863	0.1436	0.1225	0.1088	0.0970
MKDIP-E	0.1589	0.1293	0.1126	0.1019	0.0911
MKDIP-D	0.1591	0.1293	0.1126	0.1020	0.0912
MKDIP-R	**0.1563**	**0.1283**	**0.1118**	**0.1012**	**0.0907**

(b) $c = 0.5$, estimated precision factor

Method/ n	30	60	90	120	150
Jeffreys'	0.2155	0.1578	0.1300	0.1134	0.1010
RMEP	0.1761	**0.1381**	**0.1177**	0.1032	**0.0943**
RMDIP	0.1761	**0.1381**	**0.1177**	0.1032	**0.0943**
REMLP	0.2060	0.1607	0.1315	0.1120	0.1019
MKDIP-E	0.1760	**0.1381**	**0.1177**	**0.1031**	**0.0943**
MKDIP-D	0.1761	**0.1381**	**0.1177**	0.1032	**0.0943**
MKDIP-R	**0.1742**	0.1392	0.1184	0.1036	0.0949

(c) $c = 0.6$, optimal precision factor

Method/ n	30	60	90	120	150
Jeffreys'	0.2183	0.1595	0.1322	0.1146	0.1027
RMEP	0.1628	0.1332	0.1154	0.1039	0.0946
RMDIP	0.1628	0.1333	0.1154	0.1039	0.0947
REMLP	0.1867	0.1471	0.1247	0.1101	0.0990
MKDIP-E	0.1627	0.1332	0.1154	0.1038	0.0946
MKDIP-D	0.1628	0.1332	0.1154	0.1039	0.0946
MKDIP-R	**0.1598**	**0.1317**	**0.1144**	**0.1032**	**0.0940**

(d) $c = 0.6$, estimated precision factor

Method/ n	30	60	90	120	150
Jeffreys'	0.2183	0.1595	0.1322	0.1146	0.1027
RMEP	0.1805	**0.1408**	0.1201	**0.1061**	**0.0961**
RMDIP	0.1805	**0.1408**	0.1201	**0.1061**	**0.0961**
REMLP	0.2065	0.1635	0.1346	0.1166	0.1036
MKDIP-E	**0.1804**	**0.1408**	**0.1200**	**0.1061**	**0.0961**
MKDIP-D	0.1805	**0.1408**	0.1201	**0.1061**	**0.0961**
MKDIP-R	0.1814	0.1421	0.1207	0.1065	0.0965

Table 2.7: Expected difference between the true model (for TP53 network) and estimated posterior probability masses. Optimal precision factor (left) and estimated precision factor (right), with $c = 0.5$, and $c = 0.6$. The lowest distance for each sample size is written in bold.

(a) $c = 0.5$, optimal precision factor

Method/ n	15	30	45	60	75
Jeffreys'	0.2285	0.1716	0.1429	0.1242	0.1114
RMEP	0.1427	0.1165	0.1051	0.0934	0.0880
RMDIP	0.1424	0.1163	0.1048	0.0932	**0.0878**
REMLP	0.1698	0.1337	0.1199	0.1091	0.0985
MKDIP-E	0.1412	0.1161	0.1050	0.0933	0.0880
MKDIP-D	**0.1407**	**0.1158**	**0.1047**	**0.0931**	**0.0878**
MKDIP-R	0.1564	0.1247	0.1118	0.1031	0.0930

(b) $c = 0.5$, estimated precision factor

Method/ n	15	30	45	60	75
Jeffreys'	0.2285	0.1716	0.1429	0.1242	0.1114
RMEP	0.2218	0.1578	**0.1280**	0.1095	**0.0981**
RMDIP	0.2217	0.1575	0.1281	**0.1094**	**0.0981**
REMLP	0.1845	0.1505	0.1366	0.1235	0.1133
MKDIP-E	0.2149	0.1565	0.1282	0.1096	**0.0981**
MKDIP-D	0.2149	0.1564	0.1281	0.1096	**0.0981**
MKDIP-R	**0.1733**	**0.1410**	0.1281	0.1171	0.1082

(c) $c = 0.6$, optimal precision factor

Method/ n	15	30	45	60	75
Jeffreys'	0.2319	0.1723	0.1438	0.1262	0.1137
RMEP	0.1476	0.1222	0.1090	0.0987	0.0923
RMDIP	0.1474	0.1220	0.1087	0.0985	0.0921
REMLP	0.1751	0.1332	0.1192	0.1077	0.0980
MKDIP-E	0.1457	0.1215	0.1086	0.0985	0.0922
MKDIP-D	**0.1452**	**0.1211**	**0.1084**	**0.0983**	**0.0920**
MKDIP-R	0.1574	0.1217	0.1093	0.1010	0.0926

(d) $c = 0.6$, estimated precision factor

Method/ n	15	30	45	60	75
Jeffreys'	0.2319	0.1723	0.1438	0.1262	0.1137
RMEP	0.2182	0.1599	0.1304	**0.1144**	0.1032
RMDIP	0.2179	0.1597	0.1303	**0.1144**	**0.1031**
REMLP	0.1937	0.1522	0.1363	0.1235	0.1144
MKDIP-E	0.2165	0.1586	0.1304	0.1147	0.1036
MKDIP-D	0.2164	0.1585	0.1303	0.1147	0.1035
MKDIP-R	**0.1758**	**0.1418**	**0.1274**	0.1158	0.1086

2.3.5 Performance Comparison in Mixture Setup

The performance of the OBC with different prior construction methods, including OBC with the Jeffreys' prior, OBC with prior constructions methods of [40] (RMEP, RMDIP, REMLP), and OBC with the general framework of constraints (MKDIP-E, MKDIP-D, MKDIP-R), are further compared in the mixture setup with missing labels, for both the mammalian cell-cycle and the TP53 systems. Also, the OBC with prior distribution centered on the true parameters with a relatively low variance (hereinafter abbreviated as PDCOTP method in Tables 2.8 and 2.9) is considered as the comparison baseline, though it is not a practical method. Similar to the classification problems, we assume that only two components (classes) exist, normal and mutated (cancerous). Here, c_0 is fixed at 0.6 ($c_1 = 1 - c_0 = 0.4$), but the sampling is not stratified. The component-conditional SSDs (bin probabilities) for the two components are as before in the classification problem, i.e. the same as the class-conditional SSDs in the classification problem.

For each sample point, first the label (y) is generated from a Bernoulli distribution with success probability c_1, and then the bin observation is generated given the label, from the corresponding class-conditional SSD (class conditional bin probabilities vector, $\boldsymbol{p}^y$), i.e. the bin observation is a sample from a categorical distribution with parameter vector $\boldsymbol{p}^y$ but the label is hidden for the inference chain and classifier training. n sample points are generated and fed into the Gibbs inference chain with different priors from the different prior construction methods. Then the OBC is calculated based on (2.9). For each sample size, 400 Monte Carlo repetitions are done to calculate the expected true error and the error of classifying the unlabeled observed data used for the inference itself.

To have a fair comparison of different methods' class-conditional prior probability construction, we assume that we have a rough idea of the mixture weights (class probabilities). In practice this can come from existing population statistics. That is, the Dirichlet prior distribution over the mixture weights (class probabilities) parameters, $\boldsymbol{\phi}$ in $\mathcal{D}(\boldsymbol{\phi})$, are sampled in each iteration from a uniform distribution that is centered on the true mixture weights vector $+/-10\%$ interval, and fixed for all the methods in that repetition. For the REMLP and MKDIP-R that need labeled data

in their prior construction procedure, the predicted labels from using the Jeffreys' prior are used and one fourth of the data points are used in prior construction for these two methods, and all for inference. The reason for using a larger number of data points in prior construction within the mixture setup compared to the classification setup is that in the mixture setup, data points are missing their true class labels, and the initial label estimates may be inaccurate. One can use a relatively larger number of data points in prior construction, which still avoids overfitting. The regularization parameters λ_1 and λ_2 are set as in the classification problem. Optimal precision factors are used for all prior construction methods. The results are shown in Tables 2.8 and 2.9 for the mammalian cell-cycle and TP53 models, respectively. The best performance (lowest error) for each sample size and the best performance among practical methods (all other than PDCOTP), if different, is written in bold. As can be seen from the tables, in most cases the MKDIP methods have the best performance among the practical methods. With larger sample sizes, MKDIP-R even outperforms PDCOTP in the mammalian cell-cycle system.

Table 2.8: Expected errors of different Bayesian classification rules in the mixture model for the mammalian cell-cycle network. Expected true error (left) and expected error on unlabeled training data (right), with $c_0 = 0.6$. The lowest error for each sample size and the lowest error among practical methods is written in bold.

Method/ n	30	60	90	120	150
PDCOTP	**0.3216**	**0.3246**	**0.3280**	0.3309	0.3334
Jeffreys'	0.4709	0.4743	0.4704	0.4675	0.4654
RMEP	0.3417	0.3340	0.3307	0.3300	0.3299
RMDIP	0.3408	0.3336	0.3300	0.3305	0.3301
REMLP	0.3754	0.3835	0.3882	0.3857	0.3844
MKDIP-E	0.3411	0.3341	**0.3297**	0.3297	0.3306
MKDIP-D	**0.3407**	**0.3330**	0.3306	0.3304	0.3303
MKDIP-R	0.3457	0.3342	0.3299	**0.3286**	**0.3289**

Method/ n	30	60	90	120	150
PDCOTP	**0.3236**	**0.3270**	**0.3314**	0.3355	0.3339
Jeffreys'	0.4751	0.4621	0.4681	0.4700	0.4645
RMEP	0.3447	0.3409	0.3366	0.3323	0.3316
RMDIP	**0.3442**	0.3404	0.3342	0.3344	0.3343
REMLP	0.3748	0.3821	0.3908	0.3826	0.3812
MKDIP-E	0.3457	0.3386	0.3351	0.3312	0.3320
MKDIP-D	0.3482	0.3387	0.3381	0.3342	0.3334
MKDIP-R	0.3449	**0.3343**	**0.3330**	**0.3306**	**0.3275**

2.3.6 Performance Comparison on a Real Data Set

In this section the performance of the proposed methods are examined on a publicly available gene expression dataset. Here, we have considered the classification of two subtypes of non-small

Table 2.9: Expected errors of different Bayesian classification rules in the mixture model for the TP53 network. Expected true error (left) and expected error on unlabeled training data (right), with $c_0 = 0.6$. The lowest error for each sample size and the lowest error among practical methods is written in bold.

Method/ n	15	30	45	60	75
PDCOTP	**0.2746**	**0.2824**	**0.2829**	**0.2996**	**0.2960**
Jeffreys'	0.4204	0.4324	0.4335	0.4432	0.4361
RMEP	**0.3274**	**0.3204**	0.3327	**0.3402**	0.3422
RMDIP	0.3297	0.3260	0.3327	0.3406	0.3432
REMLP	0.3637	0.3687	0.3706	0.3658	0.3653
MKDIP-E	0.3312	0.3246	0.3322	0.3428	0.3386
MKDIP-D	0.3321	**0.3204**	**0.3306**	0.3436	**0.3366**
MKDIP-R	0.3872	0.3749	0.3667	0.3607	0.3586

Method/ n	15	30	45	60	75
PDCOTP	**0.2762**	**0.2818**	**0.2900**	**0.3027**	**0.2900**
Jeffreys'	0.4220	0.4314	0.4381	0.4419	0.4348
RMEP	**0.3471**	0.3350	0.3487	0.3543	0.3529
RMDIP	0.3504	0.3423	0.3496	0.3551	0.3545
REMLP	0.3489	0.3579	0.3709	0.3593	0.3556
MKDIP-E	0.3502	0.3378	0.3486	0.3585	0.3492
MKDIP-D	0.3551	**0.3329**	**0.3473**	0.3570	0.3475
MKDIP-R	0.3613	0.3583	0.3589	**0.3539**	**0.3462**

cell lung cancer (NSCLC), lung adenocarcinoma (LUA) versus lung squamous cell carcinoma (LUS). Lung cancer is the second most commonly diagnosed cancer and the leading cause of cancer death in both men and women in the United States [60]. About 84% of lung cancers are NSCLC [60] and LUA and LUS combined account for about 70% of lung cancers based on the American Cancer Society statistics for NSCLC. We have downloaded LUA and LUS datasets (both labeled as TCGA provisional) in the form of mRNA expression z-scores (based on RNA-Seq profiling) from the public database cBioPortal [61, 62] for the patient sets tagged as "All Complete Tumors", denoting the set of all tumor samples that have mRNA and sequencing data. The two datasets for LUA and LUS consist of 230 and 177 sample points, respectively. We have quantized the data into binary levels based on the following preprocessing steps. First, to remove the bias for each patient, each patient's data are normalized by the mean of the z-scores of a randomly selected subset from the list of the recurrently mutated genes (half the size of the list) from the MutSig [63] (directly provided by cBioPortal). Then, a two component Gaussian mixture model is fit to each gene in each data set, and the normalized data are quantized by being assigned to one component, namely 0 or 1 (1 being the component with higher mean). We confine the feature set to {EGFR,PIK3CA,AKT,KRAS,RAF1,BAD,P53,BCL2} which are among the genes in the most relevant signaling pathways to the NSCLC [1]. These genes are altered, in different forms, in 86% and 89% of the sequenced LUA and LUS tumor samples on the cBioPortal, respectively. There

are 256 bins in this classification setting, since the feature set consists of 8 genes. The pathways relevant to the NSCLC classification problem considered here are collected from KEGG [64, 65] Pathways for NSCLC and PI3K-AKT signaling pathways, and also from [1], as shown in Figure 2.4. The corresponding regulating functions are shown in Table 2.10.

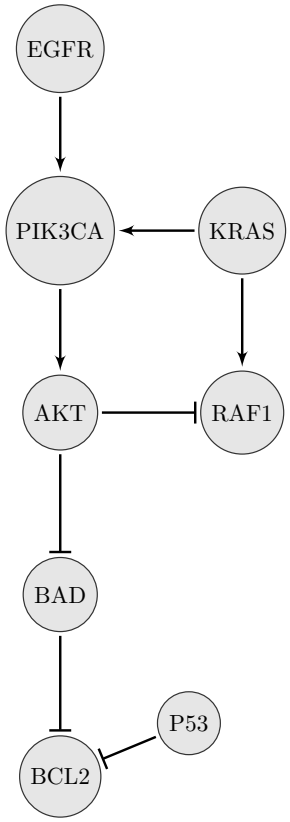


Figure 2.4: Signaling pathways corresponding to NSCLC classification. The pathways are collected from KEGG Pathways for NSCLC and PI3K-AKT pathways, and from [1].

The informative prior construction methods utilize the knowledge in the pathways in Figure 2.4, and the MKDIP methods also use the regulating relationships in Table 2.10 in order to construct prior distributions. The incidence rate of the two subtypes, LUA and LUS, varies based on demographic factors. Here, we approximate the class probability $c = P(Y = \text{LUA})$ as $c \approx 0.57$, based on the latest statistics of the American Cancer Society for NSCLC, and also based on a

Table 2.10: Regulating functions corresponding to the signaling pathways in Figure 2.4. In the Boolean functions {AND, OR, NOT} $= \{\wedge, \vee, -\}$.

Gene	Node name	Boolean regulating function
EGFR	v_1	-
PIK3CA	v_2	$v_1 \vee v_4$
AKT	v_3	v_2
KRAS	v_4	-
RAF1	v_5	$v_4 \wedge \overline{v_3}$
BAD	v_6	$\overline{v_3}$
P53	v_7	-
BCL2	v_8	$\overline{v_6} \vee \overline{v_7}$

weighted average of the rates for 11 countries given in [66]. In each Monte Carlo repetition, n sample points by stratified sampling, i.e. $n_0 = \lceil cn \rceil$ and $n_1 = n - n_0$ sample points, are randomly taken from preprocessed LUA (class 0) and LUS (class 1) datasets, respectively, for prior construction (if used by the method) and classifier training, and the rest of the sample points are held out for error estimation. For each n, 400 Monte Carlo repetitions are done to calculate the expected classification error. In the prior construction methods, first the optimization is solved for both classes with the precision factors $\alpha_0^y = 200, y \in \{0, 1\}$, and then their optimal values are estimated using the training points. For REMLP and MKDIP-R, which use a fraction of the training data in their prior construction procedure, $\min\big(20, \max(6, \lfloor 0.25 n_y \rfloor)\big)$ sample points from the training data of each class ($y \in \{0, 1\}$) are used for prior construction, and all the training data are used for inference. The regularization parameters λ_1 and λ_2 are set to 0.5 and 0.25, respectively. The results are shown in Table 2.11. In the table, the best performance among Hist, CART, RF and SVM is shown as Best Non Bayesian method. Best RM represents the best performance among RMEP, RMDIP, and REMLP. Best MKDIP denotes the best performance among the MKDIP methods. The best performing rule for each sample size is written in bold. As can be seen from the table, OBC with MKDIP prior construction methods has the best performance among the classification rules. It is also clear that the classification performance can be significantly improved when pathway prior

Table 2.11: Expected error of different classification rules calculated on a real dataset. The classification is between LUA (class 0) and LUS (class 1), with $c = 0.57$.

Method/ n	34	74	114	134	174
Best Non Bayesian	0.1764	0.1574	0.1473	0.1426	0.1371
Jeffreys'	0.1766	0.1574	0.1476	0.1425	0.1371
Best RM	0.1426	0.1289	0.1164	0.1083	0.1000
Best MKDIP	**0.1401**	**0.1273**	**0.1162**	**0.1075**	**0.0998**

knowledge is integrated for constructing prior probabilities, especially when the sample size is small.

3. CONSTRUCTING PATHWAY-BASED PRIORS WITHIN A GAUSSIAN MIXTURE MODEL FOR BAYESIAN REGRESSION AND CLASSIFICATION *

3.1 Introduction

Gaussian mixtures are useful for modeling heterogeneous populations, where the mixing proportions (probabilities) are often unknown (or subject to uncertainty). In phenotype classification or biomarker estimation problems, each component represents one sub-population in the tumor type under study and the mixing probabilities reflect the relative abundance of each tumor sub-type within the population. Given the prevalence of model uncertainty in genomic studies, a Bayesian approach is often the only course possible. In this Chapter, we continue our work on prior construction and extend it to Gaussian mixtures for Bayesian classification and regression. Here, we construct a prior distribution on an uncertainty class, in particular, a prior probability on the covariance matrix in each component in a GMM. Bayesian perspectives on (finite) Gaussian mixture models (GMMs) have been widely studied [67, 68]. We propose a rigorous framework to construct priors for a Bayesian GMM when the prior information is extracted from a set of biological signaling pathways.

This Chapter mainly addresses the following important question: Given our state of knowledge, for example, in the form of molecular interaction networks, where the underlying population is known to be a mixture of Gaussians, how can we effectively perform optimal Bayesian regression/classification by simultaneously constructing component-specific priors along with the regression/classification? In answering this question, as opposed to other Bayesian regression methods for mixture models, the optimal Bayesian regression method in this Chapter not only yields the optimal operator (and not merely the parameters) by considering the whole uncertainty class, but also finds an objective-based (optimal) prior probability that best fits our current state of knowledge.

The proposed framework consists of three major steps: (1) component assignment to each

*Reprinted with permission from S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, “Constructing Pathway-based Priors within a Gaussian Mixture Model for Bayesian Regression and Classification," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 2, pp. 524–537, 2017. Copyright 2017 IEEE.

data point, (2) prior construction, and (3) prior update via Bayesian sampling. Step (2) can be decomposed into two parts: (2a) pathway information quantification: knowledge in the biological pathways is quantified via an information theoretic formulation; and (2b) optimization: combining the data for prior construction with prior knowledge, build an objective function which is shown to be convex for the Gaussian-Wishart prior on unknown mean and precision matrix.

Throughout the Chapter we use $U(a, b)$ and $Ber(p)$ to denote the uniform distribution (with support $[a, b]$) and the Bernoulli distribution (with success probability p), respectively. $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ denotes the multivariate Gaussian (Normal) distribution with the mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$ (precision matrix $\boldsymbol{\Sigma}^{-1}$). $\mathcal{D}ir(\boldsymbol{\alpha})$ denotes the Dirichlet distribution with the parameter vector $\boldsymbol{\alpha}$. $\mathcal{W}^{-1}(\boldsymbol{\Psi}, \kappa)$ ($\mathcal{W}(\boldsymbol{\Psi}, \kappa)$) is used to represent the inverse Wishart (Wishart) distribution with the scale matrix $\boldsymbol{\Psi}$ and degree of freedom κ.

3.2 Methods

3.2.1 Optimal Bayesian Regression and Classification for a Gaussian Mixture Model

A finite Gaussian mixture model (GMM) can be written in general as

$$f(\boldsymbol{x}, y) = \sum_{i=1}^{k} p_i f_i(\boldsymbol{x}, y), \tag{3.1}$$

where each $f_i(\boldsymbol{x}, y)$, called a *mixture component*, is a Gaussian density, meaning that within each component, $(\boldsymbol{x}, y)$ has a joint Gaussian distribution (e.g., $\boldsymbol{x}$ can be gene expression data and y can be a biomarker or patient outcome) parameterized by the mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$,

$$\begin{bmatrix} \boldsymbol{x} \\ y \end{bmatrix} \sim f_i(\boldsymbol{x}, y) = \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}_{i;\boldsymbol{x}} \\ \mu_{i;y} \end{bmatrix}, \boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_{i;\boldsymbol{x},\boldsymbol{x}} & \boldsymbol{\Sigma}_{i;\boldsymbol{x},y} \\ \boldsymbol{\Sigma}_{i;y,\boldsymbol{x}} & \sigma^2_{i;y} \end{bmatrix} \Big). \tag{3.2}$$

The classical linear regression paradigm applies to each component individually; that is, to find an optimal estimator of y based on observing $\boldsymbol{x}$, the conditional density of y given $\boldsymbol{x}$, i.e.

$f_i(y|\boldsymbol{x})$, is also Gaussian [69] and hence, the optimal regression function of y that minimizes the Mean-Square Error (MSE) is a linear function of $\boldsymbol{x}$:

$$\hat{y}_i(\boldsymbol{x}) = E_{f_i}[y|\boldsymbol{x}] = \mu_{i;y} + \boldsymbol{\Sigma}_{i;\boldsymbol{x},y}^T \boldsymbol{\Sigma}_{i;\boldsymbol{x},\boldsymbol{x}}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{i;\boldsymbol{x}}). \tag{3.3}$$

These results readily extend to GMMs. For a GMM the marginal distribution of $\boldsymbol{x}$ and the conditional distribution of y given $\boldsymbol{x}$ can be written as

$$f(\boldsymbol{x}) = \sum_{i=1}^{k} p_i f_i(\boldsymbol{x}) = \sum_{i=1}^{k} p_i \mathcal{N}(\boldsymbol{\mu}_{i;\boldsymbol{x}}, \boldsymbol{\Sigma}_{i;\boldsymbol{x},\boldsymbol{x}}), \tag{3.4}$$

$$f(y|\boldsymbol{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\hat{y}_i(\boldsymbol{x}), \sigma_{i;y|x}^2), \tag{3.5}$$

where $f_i(\boldsymbol{x})$ is the marginal distribution of $\boldsymbol{x}$ for component i and

$$w_i(\boldsymbol{x}) = \frac{p_i f_i(\boldsymbol{x})}{\sum_{j=1}^{k} p_j f_j(\boldsymbol{x})}, \tag{3.6}$$

$$\sigma_{i;y|x}^2 = \sigma_{i;y}^2 - \boldsymbol{\Sigma}_{i;\boldsymbol{x},y}^T \boldsymbol{\Sigma}_{i;\boldsymbol{x},\boldsymbol{x}}^{-1} \boldsymbol{\Sigma}_{i;\boldsymbol{x},y}. \tag{3.7}$$

Thus, given full knowledge of the GMM, regressing on $\boldsymbol{x}$, the predictor of y is [70]:

$$\hat{y}(\boldsymbol{x}) = \sum_{j=1}^{k} w_j(\boldsymbol{x}) \hat{y}_j(\boldsymbol{x}). \tag{3.8}$$

For classification using a mixture model and full knowledge of parameter values, one is given a new data point $(\boldsymbol{x}_t, y_t)$ to classify. The weighing function changes to

$$w_j(\boldsymbol{x}_t, y_t) = \frac{p_j f_j(\boldsymbol{x}_t, y_t)}{\sum_{i=1}^{k} p_i f_i(\boldsymbol{x}_t, y_t)}, \tag{3.9}$$

and the output of the classification can be either soft decision (based on the weights) or hard decision ($\arg\max_j w_j(\boldsymbol{x}_t, y_t)$). The weight in (3.9) is basically the conditional probability of the

data point $(\boldsymbol{x}_t, y_t)$ belonging to component j given the parameters values.

When there is uncertainty regarding system parameters, *optimal Bayesian regression (OBR)* utilizes a prior probability distribution $\pi(\boldsymbol{\theta})$ governing the parameters of the underlying probability distribution. Following observations, the prior is updated to a posterior probability distribution $\pi^*(\boldsymbol{\theta})$ and the problem is to predict a random variable Y based on observation of predictor random vector $\boldsymbol{X}$ by a measurable function $g(\boldsymbol{X})$ that minimizes the expected MSE [71]:

$$\text{MSE} = E_{\pi^*}\Big[E_Y\big[|g(\boldsymbol{X}) - Y|^2 | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}\big]\Big]. \tag{3.10}$$

Based on the classical MSE theory, the OBR is given by

$$g_{\text{OBR}}(\boldsymbol{x}) = E_{\pi^*}[\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})], \tag{3.11}$$

where $\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})$ denotes the optimal regression for the parameterization $\boldsymbol{\theta} = [\boldsymbol{p}, \boldsymbol{\mu}, \boldsymbol{\Sigma}]$, where $\boldsymbol{p}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the collection of p_i, $\boldsymbol{\mu_i}$ and $\boldsymbol{\Sigma_i}$ of all components ($i = 1, ..., k$), respectively. Hence,

$$\hat{y}^{\text{OBR}}(\boldsymbol{x}) = \int_{\boldsymbol{\Theta}} \hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x}) \pi^*(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{3.12}$$

It can be readily seen from equation (3.12) that the OBR on the mixture model yields a nonlinear functional relation between the target and predictors:

$$\hat{y}^{\text{OBR}}(\boldsymbol{x}) = \int_{\boldsymbol{\Theta}} \sum_{j=1}^{k} \frac{p_j f_j(\boldsymbol{x})}{\sum_{i=1}^{k} p_i f_i(\boldsymbol{x})} \Big[\mu_{j;y} + \boldsymbol{\Sigma}_{j;\boldsymbol{x},y}^T \boldsymbol{\Sigma}_{j;\boldsymbol{x},\boldsymbol{x}}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{j;\boldsymbol{x}})\Big] \pi^*(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{3.13}$$

Unlike the Gaussian case investigated in detail in [71], there are no closed-form solutions for the prior update owing to the missing component labels of the data. MCMC (Markov Chain Monte Carlo) is widely used for calculating the posterior [46].

For classification under uncertainty, optimal Bayesian classification can be used for both binary classification [43] and multi-class classification [72]. Assuming that classification of a data point

into any class other than the correct class has the same loss value (zero-one loss), the Bayesian Conditional Risk Estimator [72] for classifying a complete data point $(\boldsymbol{x}_t, y_t)$ to class (component) c is equal to

$$\hat{R}(c, (\boldsymbol{x}_t, y_t)) = \sum_{i=1, i\neq c}^{k} \int_{\boldsymbol{\Theta}} w_i(\boldsymbol{x}_t, y_t|\boldsymbol{\theta})\pi^*(\boldsymbol{\theta})d\boldsymbol{\theta} = \frac{\sum_{i=1, i\neq c}^{k} p_i^{\text{eff}} f_i^{\text{eff}}(\boldsymbol{x}_t, y_t)}{\sum_{i=1}^{k} p_i^{\text{eff}} f_i^{\text{eff}}(\boldsymbol{x}_t, y_t)}, \tag{3.14}$$

where p_i^{eff} and $f_i^{\text{eff}}(\boldsymbol{x}_t, y_t)$ are the posterior expectations of the component i probability and likelihood of component i respectively, i.e.,

$$\begin{aligned} p_i^{\text{eff}} &= \int_{\boldsymbol{\Theta}} p_i \pi^*(\boldsymbol{\theta})d\boldsymbol{\theta}, \\ f_i^{\text{eff}}(\boldsymbol{x}_t, y_t) &= \int_{\boldsymbol{\Theta}} f_i(\boldsymbol{x}_t, y_t|\boldsymbol{\theta})\pi^*(\boldsymbol{\theta})d\boldsymbol{\theta}, \end{aligned} \tag{3.15}$$

and they are referred to as the *effective component probability* and *effective likelihood*, respectively. The optimal Bayesian classifier (OBC) is [72]

$$\begin{aligned} \psi_{\text{OBC}}(\boldsymbol{x}_t, y_t) &= \arg\min_{c\in\{1,\ldots,k\}} \hat{R}(c, (\boldsymbol{x}_t, y_t)) = \arg\max_{c\in\{1,\ldots,k\}} \int_{\boldsymbol{\Theta}} w_c(\boldsymbol{x}_t, y_t|\boldsymbol{\theta})\pi^*(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \arg\max_{c\in\{1,\ldots,k\}} \frac{p_c^{\text{eff}} f_c^{\text{eff}}(\boldsymbol{x}_t, y_t)}{\sum_{i=1}^{k} p_i^{\text{eff}} f_i^{\text{eff}}(\boldsymbol{x}_t, y_t)}. \end{aligned} \tag{3.16}$$

By comparing (3.16) with (3.9) one sees that the classification rule is the same except that for the OBC the effective component probabilities and component conditional effective likelihoods are used.

As explained in [71], in classical Bayesian linear regression, the connection of the regression function and prior assumptions with the underlying physical system is not specified. The same holds for classical Bayesian classification. Thus, there is a “scientific gap" in constructing functional models and making prior assumptions on model parameters when the actual uncertainty applies to the underlying system. In optimal Bayesian regression/classification, the prior distribution is placed directly on the system itself, which is the approach taken in [43], [71], [73] and here.

Table 3.1: Conjugate Prior for Gaussian Mixture

Prior on Covariance Matrix	Prior on Precision Matrix
$\boldsymbol{\mu}_i \vert \boldsymbol{\Sigma}_i \sim \mathcal{N}(\boldsymbol{m}_i, \boldsymbol{\Sigma}_i/\nu_i)$	$\boldsymbol{\mu}_i \vert \boldsymbol{\Lambda}_i \sim \mathcal{N}(\boldsymbol{m}_i, (\boldsymbol{\Lambda}_i \nu_i)^{-1})$
$\boldsymbol{\Sigma}_i \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}_i, \kappa_i)$	$\boldsymbol{\Lambda}_i \sim \mathcal{W}(\boldsymbol{W}_i, \kappa_i)$
$(p_1, \ldots, p_k) \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_k)$	$(p_1, \ldots, p_k) \sim \mathcal{D}ir(\alpha_1, \ldots, \alpha_k)$

With GMMs, unlike usual Gaussian classification, the training data are missing their true component (class) labels. Thus, closed-form calculation of the posterior probability of parameters is impossible, let alone obtaining closed forms for the effective component probabilities and component conditional effective likelihoods. One could use MCMC for numerical approximations; however, one can also use the plug-in classification rule, where point estimates of the parameters are used for classification purposes. The Bayesian posterior mean of the parameters provides decent estimation for the true parameter values. While the Bayesian posterior mean is the optimal MSE estimator, it is suboptimal for classification under Bayesian assumptions. Nevertheless, it is reasonable for comparing Bayesian classification results with frequentist classification results when only point estimates of parameters are available. Plugging in point estimates of the GMM parameters gives the following classification rule:

$$\psi_{PE}(\boldsymbol{x}_t, y_t) = \arg\max_{c \in \{1,\ldots,k\}} \frac{\hat{p}_c f_c(\boldsymbol{x}_t, y_t | \hat{\boldsymbol{\theta}})}{\sum_{i=1}^{k} \hat{p}_i f_i(\boldsymbol{x}_t, y_t | \hat{\boldsymbol{\theta}})}. \tag{3.17}$$

3.2.1.1 Conjugate priors for Gaussian mixture model

Considering the conjugate prior for the GMM, one would have the structure summarized in Table 3.1. There are four independent parameters that fully characterize the Gaussian-Inverse-Wishart prior probability over each component: $\boldsymbol{m}_i$, ν_i, $\boldsymbol{\Psi}_i$ ($\boldsymbol{W}_i = \boldsymbol{\Psi}_i^{-1}$), and κ_i. Two of these parameters, ν_i and κ_i, are scalars, regardless of the dimension d of the problem. These two parameters determine the spread of the prior: increasing ν_i or κ_i leads to shrinkage in our uncertainty

regarding the mean or covariance matrix (precision matrix), respectively. The matrix $\boldsymbol{\Psi}_i$ ($\boldsymbol{W}_i$) is called the *scale-matrix* of the inverse Wishart (Wishart) distribution and determines the mean of the covariance matrix (precision matrix) as

$$E[\boldsymbol{\Sigma}_i] = \frac{\boldsymbol{\Psi}_i}{\kappa_i - d - 1}, \quad (\text{or} \quad E[\boldsymbol{\Lambda}_i] = \boldsymbol{W}_i \kappa_i).$$

The Dirichlet distribution over the component probabilities is parameterized by a vector of k positive real numbers $(\alpha_1, \ldots, \alpha_k)$.

3.2.2 Regularized Expected Mean Log-Likelihood Prior

Prior knowledge is in the form of pathways. Entities in a set of pathways are denoted by $x(i)$ (as the i-th element of vector $\mathbf{x}$). An *activating pathway segment* (APS) $x(i) \longrightarrow x(j)$ means that $x(i)$ up-regulated (UR) implies $x(j)$ UR (in some time steps). A *repressing pathway segment* (RPS) $x(i) \dashv x(j)$ means that $x(i)$ UR implies $x(j)$ down-regulated (DR). A pathway is an APS/RPS sequence. If $\mathcal{G}$ is a set of pathways, then $\mathcal{G}_{\mathcal{A}}$ and $\mathcal{G}_{\mathcal{R}}$ include all APS and RPS segments in $\mathcal{G}$, respectively. The *regulatory set* R_x for gene x is the set of genes regulated by x via some APS/RPS.

Pathway information is marginal and incomplete with respect to regulation. Following [51], APS and RPS relations are specified probabilistically by

$$\begin{aligned} &\text{APS: } \mathrm{E}_{\boldsymbol{\theta}}[\Pr(x(j_a) = \text{UR} | x(i_a) = \text{UR})] = 1 - \delta_{i_a j_a}, \\ &\text{RPS: } \mathrm{E}_{\boldsymbol{\theta}}[\Pr(x(j_r) = \text{DR} | x(i_r) = \text{UR})] = 1 - \delta_{i_r j_r}, \end{aligned} \tag{3.18}$$

where the nonnegative *conditioning parameters* $\delta_{i_a j_a}$ and $\delta_{i_r j_r}$, which lie in [0,1], measure the loss of complete regulation resulting from context effects. For Gaussian joint distributions and acyclic pathways, the inequalities are changed to

$$\begin{aligned} &\text{APS: } \mathrm{E}_{\boldsymbol{\theta}}\left[\rho_{x(i_a),x(j_a)}\right] = 1 - \alpha_{i_a j_a}, \\ &\text{RPS: } \mathrm{E}_{\boldsymbol{\theta}}\left[\rho_{x(i_r),x(j_r)}\right] = -1 + \alpha_{i_r j_r}, \end{aligned} \tag{3.19}$$

where$\rho_{x(i),x(j)}$ denotes the correlation coefficient between two entities $x(i)$ and $x(j)$, $0 \leq \alpha_{i_a j_a} \leq$

1, and $0 \leq \alpha_{i_r j_r} \leq 1$.

The conditional Shannon entropy of a gene $x(i)$ given $R_{x(i)}$ is utilized via the constraint $\mathrm{E}_{\boldsymbol{\theta}}[\mathrm{H}_{\boldsymbol{\theta}}(x(i)|R_{x(i)})] = \eta_i$, where $\mathrm{H}_{\boldsymbol{\theta}}(v_1|v_2)$ is the conditional Shannon entropy obtained by a $\boldsymbol{\theta}$-parameterized distribution.

In the Regularized Expected Mean Log-Likelihood Prior (REMLP) approach, we use a measure of similarity between the true distribution ($\boldsymbol{\theta}_{true}$) and an arbitrary distribution ($\boldsymbol{\theta}$). The Kullback-Leibler (KL) divergence provides a measure of the difference:

$$
\begin{aligned}
&KL(\boldsymbol{\theta}_{true}, \boldsymbol{\theta}) = \\
&\int_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}|\boldsymbol{\theta}_{true}) \log \frac{f(\mathbf{x}|\boldsymbol{\theta}_{true})}{f(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x} = \\
&\int_{\mathbf{x}\in\mathcal{X}} [f(\mathbf{x}|\boldsymbol{\theta}_{true}) \log f(\mathbf{x}|\boldsymbol{\theta}_{true}) - f(\mathbf{x}|\boldsymbol{\theta}_{true}) \log f(\mathbf{x}|\boldsymbol{\theta})] d\mathbf{x}.
\end{aligned}
$$

Since $KL(\boldsymbol{\theta}_{true}, \boldsymbol{\theta}) \geq 0$ and $f(\mathbf{x}|\boldsymbol{\theta}_{true})$ is fixed, $KL(\boldsymbol{\theta}_{true}, \boldsymbol{\theta})$ is minimized by maximizing

$$
\begin{aligned}
\rho(\boldsymbol{\theta}_{true}, \boldsymbol{\theta}) &= \int_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}|\boldsymbol{\theta}_{true}) \log f(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x} \\
&= \mathrm{E}[\log f(\mathbf{x}|\boldsymbol{\theta})|\boldsymbol{\theta}_{true}],
\end{aligned}
\tag{3.20}
$$

which can therefore be treated as a similarity measure between $\boldsymbol{\theta}_{true}$ and $\boldsymbol{\theta}$.

Suppose the sample S_n is split into two parts for each class $y \in \{0, 1\}$: $S_{n_y^p}^{prior,y}$ and $S_{n_y^t}^{train,y}$, with $n_y = n_y^p + n_y^t$ and $n = n_0 + n_1$. Dropping the index y for notational ease, the sample set (consisting of $n_p = n_0^p$ or $n_p = n_1^p$ sample points) used for prior construction (for each class) is denoted by $S_{n_p}^{prior}$. $\rho(\boldsymbol{\theta}_{true}, \boldsymbol{\theta})$ has the sample-mean estimate

$$
\ell_{n_p}(\boldsymbol{\theta}) := \frac{1}{n_p} \ell(\boldsymbol{\theta}; S_{n_p}^{prior}) = \frac{1}{n_p} \sum_{i=1}^{n_p} \log f(\mathbf{x}_i|\boldsymbol{\theta}),
\tag{3.21}
$$

where $\ell(\boldsymbol{\theta}; S_{n_p}^{prior})$ denotes the log-likelihood function. In other words, ℓ_{n_p} in (3.21) can be interpreted as an estimator of the similarity measure in (3.20) [74],[75].

We consider the following optimization with multiple constraints in which, owing to incon-

sistencies in the prior knowledge, slack variables (ξ_i, $\varepsilon_{i_a j_a}$, $\varepsilon_{i_r j_r}$) are introduced to relax the constraints:

$$\pi_{\mathbf{REMLP}}(\boldsymbol{\theta}) := \underset{\substack{\pi(\boldsymbol{\theta}) \in \Pi, \xi_i \geq 0 \\ \varepsilon_{i_a j_a} \geq 0, \varepsilon_{i_r j_r} \geq 0}}{\arg\min} \quad -(1-\lambda_1-\lambda_2)\mathrm{E}_{\boldsymbol{\theta}}\Big[\ell_{n_p}(\boldsymbol{\theta})\Big] + \lambda_1 \sum_{i=1}^{|\mathcal{C}|} \xi_i + \lambda_2 \Big[\sum_{(i_a,j_a)\in\mathcal{G}_{\mathcal{A}}} \varepsilon_{i_a j_a} + \sum_{(i_r,j_r)\in\mathcal{G}_{\mathcal{R}}} \varepsilon_{i_r j_r} \Big] \tag{3.22}$$

subject to the following constraints:

$$\eta_i - \xi_i \leq \mathrm{E}_{\boldsymbol{\theta}}\Big[\mathrm{H}_{\boldsymbol{\theta}}(x(i)|R_{x(i)})\Big] \leq \eta_i + \xi_i, x(i) \in \mathcal{G} \tag{3.23}$$

$$1 - \delta_{i_a j_a} - \varepsilon_{i_a j_a} \leq \mathrm{E}_{\boldsymbol{\theta}}\Big[\Pr(x(j_a) = \mathrm{UR}|x(i_a) = \mathrm{UR})\Big] \leq 1 - \delta_{i_a j_a} + \varepsilon_{i_a j_a}, (i_a, j_a) \in \mathcal{G}_{\mathcal{A}} \tag{3.24}$$

$$1 - \delta_{i_r j_r} - \varepsilon_{i_r j_r} \leq \mathrm{E}_{\boldsymbol{\theta}}\Big[\Pr(x(j_r) = \mathrm{DR}|x(i_r) = \mathrm{UR})\Big] \leq 1 - \delta_{i_r j_r} + \varepsilon_{i_r j_r}, (i_r, j_r) \in \mathcal{G}_{\mathcal{R}}, \tag{3.25}$$

where Π is the feasible prior region and $\lambda_1, \lambda_2 \geq 0$, with $\lambda_1 + \lambda_2 \leq 1$, are regularization parameters. Assuming Gaussian distributions, the APS/RPS equations become

$$1 - \alpha_{i_a j_a} - \varepsilon_{i_a j_a} \leq \mathrm{E}_{\boldsymbol{\theta}}\Big[\rho_{x(i_a),x(j_a)}\Big] \leq 1 - \alpha_{i_a j_a} + \varepsilon_{i_a j_a}, (i_a, j_a) \in \mathcal{G}_{\mathcal{A}} \tag{3.26}$$

$$-1 + \alpha_{i_r j_r} - \varepsilon_{i_r j_r} \leq \mathrm{E}_{\boldsymbol{\theta}}\Big[\rho_{x(i_r),x(j_r)}\Big] \leq -1 + \alpha_{i_r j_r} + \varepsilon_{i_r j_r}, (i_r, j_r) \in \mathcal{G}_{\mathcal{R}}. \tag{3.27}$$

In the sequel of this Chapter we assume complete information so that the conditioning parameters are 0, the Shannon entropy is 0, and in the Gaussian case the correlation is 1 for APS and -1 for RPS.

3.2.2.1 *Review of REMLP method for multivariate Gaussian with normal-Wishart prior distributions*

In this subsection, an overview of the REMLP method is provided for the multivariate Gaussian with Normal-Wishart prior distribution. The optimization framework in (3.22)-(3.25) can be decomposed into two convex optimization problems for the multivariate Gaussian model with a Gaussian-Wishart prior distribution, to employ existing methods for solving convex problems. In the first problem, λ_2 is set to 0, so that only the information in the regulatory set constraints is used. Solving it with respect to $\mathbf{m}$ yields $\mathbf{m} = \frac{1}{n_p}\sum_{i=1}^{n_p}\mathbf{x}_i$. If we assume that there is only one regulatory set constraint for one gene x, then the precision matrix and the scale matrix of the Wishart distribution governing it can be represented in block format as

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{R_x} & \mathbf{\Lambda}_{12} & \mathbf{\Lambda}_{13} \\ \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{x} & \mathbf{\Lambda}_{23} \\ \mathbf{\Lambda}_{31} & \mathbf{\Lambda}_{32} & \mathbf{\Lambda}_{33} \end{bmatrix}; \mathbf{W} = \begin{bmatrix} \mathbf{W}_{R_x} & \mathbf{W}_{12} & \mathbf{W}_{13} \\ \mathbf{W}_{21} & \mathbf{W}_{x} & \mathbf{W}_{23} \\ \mathbf{W}_{31} & \mathbf{W}_{32} & \mathbf{W}_{33} \end{bmatrix}. \tag{3.28}$$

Since

$$\begin{aligned} \mathbf{\Lambda}_x - \mathbf{\Lambda}_{23}\mathbf{\Lambda}_{33}^{-1}\mathbf{\Lambda}_{32} \sim \mathcal{W}(\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}, \\ \kappa - \dim(\mathbf{W}_{33})), \end{aligned} \tag{3.29}$$

the optimization in (3.22) with the constraint in (3.23) can be restated as

$$\min_{\mathbf{W}>0,\xi\geq 0} -\frac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}| - \kappa \text{tr}(\mathbf{W}\mathbf{V})\Big] + \lambda_1\xi \tag{3.30}$$

$$\begin{aligned} \text{Subject to} \quad & -\log|\mathbf{W}_x - \mathbf{W}_{23}\mathbf{W}_{33}^{-1}\mathbf{W}_{32}| - \\ & \psi(\frac{\kappa - (p - |R_x| - 1)}{2}) \leq \xi; \quad \xi \geq \underline{\xi}, \end{aligned} \tag{3.31}$$

which is a convex programming [39]. In the equation above, $\mathbf{V} = \frac{1}{n_p}\sum_{i=1}^{n_p}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$, $\underline{\xi} = -log(\pi e)$, and in (3.29) $\dim(\cdot)$ returns the dimension of a matrix.

Incorporating all the entities' regulatory set constraints simultaneously, by considering the corresponding submatrix for a gene and its regulatory set, the optimization problem in (3.30)-(3.31) can be extended to the following: for any $\xi_i \geq \underline{\xi}$,

$$\min_{\mathbf{W}>0,\xi_i\geq 0} -\frac{1}{2}(1-\lambda_1)\Big[\log|\mathbf{W}| - \kappa \text{tr}(\mathbf{W}\mathbf{V})\Big] + \lambda_1 \sum_{i=1} \xi_i \tag{3.32}$$

$$\text{Subject to} \quad -\log|\overline{\mathbf{W}}_{x(i)}| - \psi\Big(\frac{\kappa-(p-|R_{x(i)}|-1)}{2}\Big) \leq \xi_i, \tag{3.33}$$

where

$$\overline{\mathbf{W}}_{x(i)} := \mathbf{W}_{x(i)} - \mathbf{W}_{x(i),\mathbf{g}\backslash\bar{R}_{x(i)}} \mathbf{W}^{-1}_{\mathbf{g}\backslash\bar{R}_{x(i)}} \mathbf{W}^{T}_{x(i),\mathbf{g}\backslash\bar{R}_{x(i)}}. \tag{3.34}$$

Here $\bar{R}_{x(i)}$ denotes the union of $x(i)$ and $R_{x(i)}$. The optimization problem in (3.32)-(3.33) can be solved by the log-barrier interior point method.

In the second optimization problem, the regulation information from the pathways, formulated as constraints in (3.26) and (3.27), are incorporated. The second optimization paradigm tries to find the closest (in terms of the Frobenius norm) positive-definite matrix $\mathbf{\Psi} = \mathbf{W}^{-1}$ to the solution of the first optimization problem in (3.32)-(3.33), $\mathbf{\Psi}^* = \mathbf{W}^{*-1}$, while satisfying the correlation coefficient constraints in (3.26) and (3.27). Since the elements of the covariance matrix are distributed according to an inverse Wishart distribution, i.e., $\mathbf{\Sigma} = [\sigma_{ij}]_{p\times p} \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \kappa)$, $\text{E}[\sigma_{ij}] = \frac{1}{k-p-1}\psi_{ij}$, for $i, j \in \{1, ..., p\}$. Hence, the expected correlations can be approximated by

$$\begin{aligned} \text{E}[\rho_{ij} = \rho_{x(i),x(j)}] = \text{E}\left[\frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}\right] &\approx \frac{\text{E}[\sigma_{ij}]}{\frac{1}{k-p-1}\sqrt{\psi^*_{ii}\psi^*_{jj}}} \\ &= \frac{\psi_{ij}}{\sqrt{\psi^*_{ii}\psi^*_{jj}}}. \end{aligned} \tag{3.35}$$

Using the approximation (3.35) for the constraints (3.26) and (3.27), and using the Frobenius norm penalty, the second optimization yields the following convex optimization problem [39],

$$\min_{\mathbf{\Psi}>0,\varepsilon_{ij}\geq 0} (1-\lambda_2)||\mathbf{\Psi} - \mathbf{\Psi}^*||_F^2 + \lambda_2\Big[\sum_{(i_a,j_a)\in\mathcal{G}_\mathcal{A}} \varepsilon_{i_a j_a} + \sum_{(i_r,j_r)\in\mathcal{G}_\mathcal{R}} \varepsilon_{i_r j_r}\Big], \tag{3.36}$$

subject to the constraints

$$\begin{cases} 1-\varepsilon_{i_a j_a} \leq \frac{\psi_{i_a j_a}}{\sqrt{\psi^*{}_{i_a i_a}\psi^*{}_{j_a j_a}}} \leq 1; \quad (i_a, j_a) \in \mathcal{G}_{\mathcal{A}} \\ 1-\varepsilon_{i_r j_r} \leq \frac{-\psi_{i_r j_r}}{\sqrt{\psi^*{}_{i_r i_r}\psi^*{}_{j_r j_r}}} \leq 1; \quad (i_r, j_r) \in \mathcal{G}_{\mathcal{R}} \end{cases}, \tag{3.37}$$

where $\lambda_2 \in (0, 1)$ is a regularization factor balancing two functions. The second optimization problem is a linearly constrained quadratic programming problem.

In summary, the general optimization problem in (3.22)-(3.25) is decomposed into two sequential optimization problems: first, the optimization in (3.32)-(3.33), and second, the optimization in (3.36)-(3.37).

3.2.3 Prior Construction and Inference for a GMM

In this section, we propose a new approach for constructing priors over the GMM and explain how it can be utilized for Bayesian regression and classification. We will show that the prior construction bundled with the prior update via Bayesian sampling results in improved inference, which results in lower regression and classification errors. Fig. 3.1 shows the steps involved in prior construction for Bayesian regression and classification for a GMM.

3.2.3.1 Step 1: Initialization using Data

In the first step of the algorithm, an initialization is made for the *latent variables (component allocations, labels)*, since such allocation data are missing. This can be done via an expert who can, to some level (possibly with some errors), label the data points. In the absence of an expert, we can use expectation maximization (EM) [76, 77] to find an initial allocation; however, the allocation should be aligned with the prior knowledge, i.e. not only do we need clustering of the data points for different components, we also need to assign each set of prior information to the clustered points. The reason for this is that the mixture likelihood is invariable under permutation of the components, but each set of prior knowledge, e.g. biological pathways or networks, corresponds to one specific component. Thus, we need an additional identifiability constraint to distinguish the component labels we get from EM. This identifiability constraint can be an inequality such as

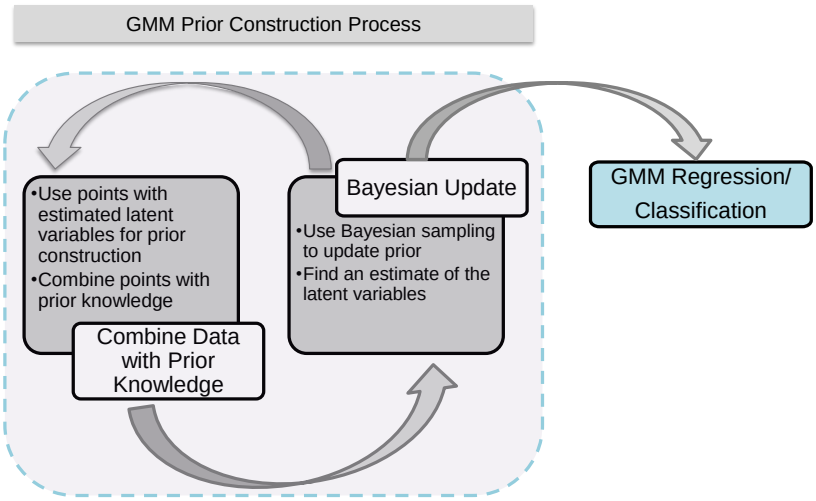
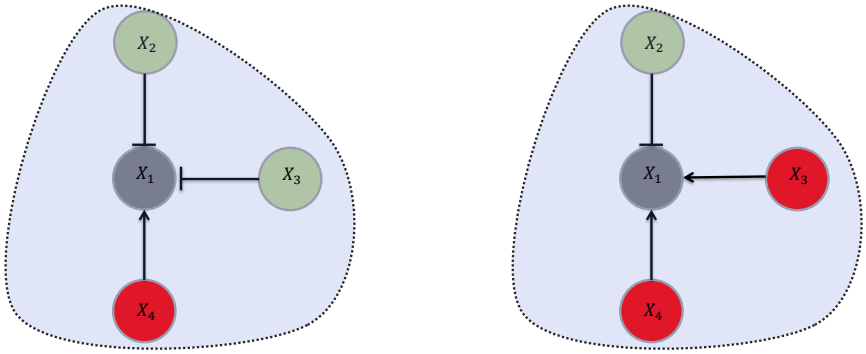


Figure 3.1: A schematic for the prior construction method.

ordering of the mean expression value of a specific entity in the pathways or ordering of component probabilities. For example, if we know that one subtype of a specific cancer is more prevalent than another subtype, then this constraint can be translated to an inequality over component probabilities (mixture weights) in the mixture model to distinguish the components.

As an illustration consider a simple toy example, where a disease has two subtypes, A and B. The data collected from patients having this disease are not labeled for the subtype, i.e. do not have the component allocations. The prior knowledge about each subtype is in the form of signaling pathways in Fig. 3.2.

As can be seen in the figure, the regulatory effect of X_3 on X_1 is different in the two subtypes: in subtype A the edge connecting X_3 to X_1 is an RPS, but in subtype B it is an APS. Also, from the domain knowledge, we know that subtype A is more prevalent. This translates to an inequality: the component probability (mixture weight) of subtype A is greater than the component probability of subtype B. Since the data are not labeled, an initial estimate of GMM parameters is calculated and an initialization is done for the latent variables by EM. Since EM is invariant under permutation of components, to align the initialization with the prior information, the component



(a) Subtype A pathway (b) Subtype B pathway

Figure 3.2: Toy example pathways.

with the higher estimated probability (estimated mixture weight) is assigned label (subtype) A and the other component is assigned label (subtype) B.

3.2.3.2 Step 2: Prior construction

In the second step of the algorithm, prior construction is done for each component based on combining the corresponding pathway information and the data according to the latent variables from the previous step. In the absence of full knowledge regarding the model parameters, any partial knowledge that can constrain the model space can be utilized to enhance the performance of the inference and prediction. To avoid increasing the computational complexity of posterior computation, the prior is confined to conjugate priors over the i-th component of the mixture model and mixture weights.

To set the mean vector and scatter matrix of each component's Gaussian-Inverse-Wishart distribution, the REMLP introduced in Section 3.2.2 is employed; however, here we propose that all data points be used for both prior construction and prior update, a similar approach to empirical Bayesian methods, instead of splitting the data into two sets for prior construction and prior update. Thus, all data points are used for prior construction and again for updating the constructed priors to get the posterior. Our reasoning is that, since the data points are missing their true class labels, and

the initial label estimates are inaccurate, by not utilizing all data points used in prior construction for prior update, one would not exploit all information in the data points.

For the hyperparameters of the Dirichlet distribution over the mixture weights, we simply set them according to the sample size and the proportion of the component allocations from the initialization step. The intuition behind this is that these hyperparameters are like the number of data points previously observed from each component; however, the initial labels are inaccurate, so the α_i are set by assuming that a fraction of the sample size is observed with accurate latent allocations. κ and ν can be viewed as the level trust in the prior construction step, that is, how much one is going to trust the initial labels. These are also set by the same intuition as the sample size with inaccurate initial labels having equivalent information to a smaller sample size with accurate labels. The size of the comparable smaller sample size with accurate labels is set based on a heuristic: as the sample size increases, the initial labels become more accurate, so that the size of the comparable smaller sample size with hundred percent accurate labels is set to a larger fraction of the sample size.

3.2.3.3 Step 3: Prior update via Bayesian sampling

In this step, the *constructed prior* is updated using the data augmentation algorithm [78]. Data augmentation is a special case of Gibbs sampling, where the parameters and missing labels are iteratively generated from their full conditional distributions, $\mathbf{z}^{(m)} \sim f(\mathbf{z}|S_n, \boldsymbol{\theta}^{(m)})$ and $\boldsymbol{\theta}^{(m+1)} \sim \pi(\boldsymbol{\theta}|S_n, \mathbf{z}^{(m)})$. Within a Gaussian-mixture-model framework, under random sampling, and by using the conjugate Gaussian-Inverse-Wishart distribution for each Gaussian component and the conjugate Dirichlet distribution for class (component) probabilities, the full conditionals take the following forms. The full conditional distribution of labels given the parameters is a Multinomial distribution, that is, for the ith data point $z_{ij}^{(m)} \sim f(z|S_n, \theta^{(m)}) = Multinomial(w_1, \ldots, w_k)$, where w_j is calculated by (3.9) using the latest sample of $\boldsymbol{\theta}^{(m)} = [\boldsymbol{p}^{(m)}, \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}]$. The conditional distribution of class (component) probabilities $(p_1, \ldots, p_k)$ conditioned on the data and $\mathbf{z}^{(m)}$ is a Dirichlet distribution with updated hyperparameters, and the conditional distribution of the mean and covariance matrix of each component is again a Gaussian-Inverse-Wishart with updated

hyperparameters. The equations for these updates are provided in Algorithm 2.

Here, for the regression problem with the test data points missing both the component labels and the regression target value, and where the objective is predicting target values, an estimate of the output is calculated in each chain iteration based on the latest sample of the parameters in the chain. This, in fact, gives a numerical approximation of (3.13), which cannot be analytically calculated. Specifically,

$$\hat{y}^{\mathrm{OBR}}(\boldsymbol{x}) \approx \sum_{m=1}^{MCIters} \sum_{j=1}^{k} \frac{p_j^{(m)} f_j^{(m)}(\boldsymbol{x})}{\sum_{i=1}^{k} p_i^{(m)} f_i^{(m)}(\boldsymbol{x})} \Big[\mu_{j;y}^{(m)} + \boldsymbol{\Sigma}_{j;\boldsymbol{x},y}^{T(m)} \boldsymbol{\Sigma}_{j;\boldsymbol{x},\boldsymbol{x}}^{-1(m)} (\boldsymbol{x} - \boldsymbol{\mu}_{j;\boldsymbol{x}}^{(m)})\Big], \tag{3.38}$$

where $MCIters$ is the number of the runs of the MCMC chain in Algorithm 2. Also, for the classification problem with the test data points missing only the component labels and where the objective is predicting labels, in each chain iteration the component weights (probability of the point belonging to each component) for the test data points are calculated based on (3.9). At the end of the chain iterations, each test data point is assigned to the component label with the highest sum of weights calculated during the chain iterations. This gives the following numerical (Monte Carlo) approximation of the OBC classification rule ((3.15) and (3.16)):

$$\psi_{\mathrm{OBC}}(\boldsymbol{x}_t, y_t) \approx \arg\max_{c \in \{1,\ldots,k\}} \sum_{m=1}^{MCIters} w_c(\boldsymbol{x}_t, y_t | \boldsymbol{\theta}^{(m)}). \tag{3.39}$$

At the end of this step, the posterior probability can be obtained from the Bayesian sampling. Also, the posterior mean of the parameters (mean of the generated samples after burn-in period and thinning) can be used as Bayesian point estimates of the parameters. These estimates can be plugged in to estimate true parameter values for suboptimal Bayesian classification in (3.17).

3.2.3.4 Step 4: Latent variable allocation and iteration

In this step the posterior mean of the parameters from the previous step is used for plug-in classification (equation (3.17)) of the unlabeled training data to get new estimates of latent variables (component labels) for the unlabeled training data. Then the method goes back to step 2, the data

according to these new latent variable estimates are combined with prior knowledge, and steps 2, 3 and 4 are iterated.

The proposed framework is summarized in Algorithms 1 and 2.

Algorithm 1 Bayesian GMM Regression/Classification bundled with Gaussian-Inverse-Wishart Prior Construction

---

Input: Pathway info, Unlabeled Training Data Points S_n, Test Point (unlabeled) $(\mathbf{x}'^t, y'^t)$, Test Point (unlabeled and missing target expression level) $\mathbf{x}^t$
Output: Posterior estimates of mean vectors, covariance matrices and mixing probabilities, target gene expression estimate $\hat{y}$ for $\mathbf{x}^t$, label estimates $\hat{z}'$ for $(\mathbf{x}'^t, y'^t)$, Hyper-parameters
Initialize: Initial latent allocations $\mathbf{z}^{(0)}$ from EM, Initial hyper-parameters: $\boldsymbol{\Psi}^{*(0)}$, $\boldsymbol{m}^{*(0)}$, $\boldsymbol{\alpha}^{*(0)}$, $\boldsymbol{\kappa}^{*(0)}$, $\boldsymbol{\nu}^{*(0)}$ and initial $\hat{y} = 0$
for $i \in 0 : \mathrm{NumIt} - 1$ **do**
  for $j = 1 : k$ **do**
    $S_j \leftarrow$ Extract points corresponding to component j from S_n according to $\mathbf{z}^{(i)}$
    $\boldsymbol{\Psi}_j^{(i+1)}$, $\boldsymbol{m}_j^{(i+1)}$, $\boldsymbol{\alpha}^{(i+1)}$, $\kappa_j^{(i+1)}$, $\nu_j^{(i+1)}$ (or $\pi_j(\boldsymbol{\theta}_j)$) $\leftarrow$ Prior Construction and solving optimization problem (3.22) with initial point of $\boldsymbol{\Psi}^{*(i)}_j$, $\boldsymbol{m}^{*(i)}_j$ and using S_j
  end for
  $\hat{y}^{(i+1)}, \hat{z}'^{(i+1)}, \mathbf{z}^{(i+1)}, \hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{p}}, \boldsymbol{\alpha}^{*(i+1)}, \boldsymbol{\Psi}^{*(i+1)}, \boldsymbol{m}^{*(i+1)}$,
  $\boldsymbol{\kappa}^{*(i+1)}, \boldsymbol{\nu}^{*(i+1)} \leftarrow$ Run Algorithm 2 with $\pi(\boldsymbol{\theta})$ ($\boldsymbol{m}^{(i+1)}$, $\boldsymbol{\alpha}^{(i+1)}$, $\boldsymbol{\Psi}^{(i+1)}$, $\boldsymbol{\kappa}^{(i+1)}$, $\boldsymbol{\nu}^{(i+1)}$), S_n (blind to initial allocations), $\mathbf{x}^t$ and $(\mathbf{x}'^t, y'^t)$
end for
return $\hat{y}^{(\mathrm{NumIt})}, \hat{z}'^{(\mathrm{NumIt})}, \mathbf{z}^{(\mathrm{NumIt})}, \hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{p}}, \boldsymbol{\alpha}^{*(\mathrm{NumIt})}$,
$\boldsymbol{\Psi}^{*(\mathrm{NumIt})}, \boldsymbol{m}^{*(\mathrm{NumIt})}, \boldsymbol{\kappa}^{*(\mathrm{NumIt})}, \boldsymbol{\nu}^{*(\mathrm{NumIt})}$

3.3 Results and Discussion

3.3.1 Simulation Setup

3.3.1.1 Synthetic pathway generation

In this section we examine the performance on synthetic pathways. Since (3.26) and (3.27) are symmetric but not directional, the method is only applied to directed acyclic pathways. The pathways are synthesized based on the following steps:

- Input parameters: Number of nodes n_{nodes}, minimum number of levels $L_{\min}$, maximum

Algorithm 2 Prior Update and Inference via Bayesian Sampling (modified from [78])

Input: Prior hyper-parameters $\boldsymbol{m}, \boldsymbol{\alpha}, \boldsymbol{\Psi}, \boldsymbol{\kappa}, \boldsymbol{\nu}$, Unlabeled training data S_n, Test data $\mathbf{x}^t$,Test data $(\mathbf{x'}^t, y'^t)$
Output: Posterior $\pi^*(\boldsymbol{\theta})$ (Posterior hyper-parameters), Posterior mean estimates of GMM parameters, $\hat{y}$, $\hat{z}^{'}$, $\mathbf{z}$
Initialize: Set all the elements in $\hat{y}, \hat{z}_j^{'}, \hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{\Psi}}, \hat{\boldsymbol{m}}, \hat{\boldsymbol{\kappa}}, \hat{\boldsymbol{\nu}}$ to zero
for $m = 1 : \text{MCIters}$ **do**
  Generate $z_{ij}^{(m)} \sim f(z|S_n, \theta^{(m-1)})$ (Multinomial distribution)
  $\mathbf{x}_j \leftarrow$ Collect all the points in component j from S^n based on $z_{ij}^{(m)}$
  $\hat{\boldsymbol{p}}^{(m)} \sim \mathcal{D}ir(\alpha_1 + \sum_{i=1}^n z_{i1}^{(m)}, \ldots, \alpha_k + \sum z_{ik}^{(m)})$
  for $j = 1 : k$ **do**
    $n_j^{(m)} \leftarrow \sum_{i=1}^n z_{ij}^{(m)}$
    $\hat{\kappa}_j^{(m)} \leftarrow \kappa_j + n_j^{(m)}$
    $\hat{\nu}_j^{(m)} \leftarrow \nu_j + n_j^{(m)}$
    $\hat{\boldsymbol{\Psi}}_j^{(m)} \leftarrow \boldsymbol{\Psi}_j + (n_j^{(m)} - 1)\boldsymbol{V}_j^{(m)} + \frac{\nu_j n_j^{(m)}}{\nu_j + n_j^{(m)}}(\hat{\mu_j} - \boldsymbol{m}_j)(\hat{\mu_j} - \boldsymbol{m}_j)^T$ ($\hat{\mu_j}$ is sample mean of $\mathbf{x}_j$, and $\boldsymbol{V}_j^{(m)}$ is sample covariance of $\mathbf{x}_j$)
    $\hat{\boldsymbol{m}}_j^{(m)} \leftarrow \frac{\nu_j \boldsymbol{m}_j + \sum_{r=1}^{n_j^{(m)}} x_j^r}{\nu_j + n_j^{(m)}}$
    Generate $\hat{\boldsymbol{\Sigma}}_j^{(m)} \sim \mathcal{W}^{-1}(\hat{\boldsymbol{\Psi}}_j^{(m)}, \hat{\kappa}_j^{(m)})$
    Generate $\hat{\boldsymbol{\mu}}_j^{(m)} \sim \mathcal{N}(\hat{\boldsymbol{m}}_j^{(m)}, \hat{\boldsymbol{\Sigma}}_j^{(m)}/\hat{\nu}_j^{(m)})$
  end for
  $\hat{y}^{(m)} \leftarrow$ Use equation (3.8), with $\hat{\boldsymbol{p}}^{(m)}$, $\hat{\boldsymbol{\mu}}^{(m)}$, $\hat{\boldsymbol{\Sigma}}^{(m)}$ and $\mathbf{x}^t$
  $\hat{y} \leftarrow \hat{y} + \hat{y}^{(m)}$
  for $j = 1 : k$ **do**
    $\hat{z}_j^{'(m)} \leftarrow$ Use equation (3.9), with $\hat{\boldsymbol{p}}^{(m)}$, $\hat{\boldsymbol{\mu}}^{(m)}$, $\hat{\boldsymbol{\Sigma}}^{(m)}$ and $(\mathbf{x'}^t, y'^t)$
    $\hat{z}_j^{'} \leftarrow \hat{z}_j^{'} + \hat{z}_j^{'(m)}$
  end for
  $\hat{\boldsymbol{\Sigma}} \leftarrow \hat{\boldsymbol{\Sigma}} + \hat{\boldsymbol{\Sigma}}^{(m)}$
  $\hat{\boldsymbol{\mu}} \leftarrow \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\mu}}^{(m)}$
  $\hat{\boldsymbol{p}} \leftarrow \hat{\boldsymbol{p}} + \hat{\boldsymbol{p}}^{(m)}$
  $\hat{\boldsymbol{\Psi}} \leftarrow \hat{\boldsymbol{\Psi}} + \hat{\boldsymbol{\Psi}}^{(m)}$
  $\hat{\boldsymbol{m}} \leftarrow \hat{\boldsymbol{m}} + \hat{\boldsymbol{m}}^{(m)}$
  $\hat{\boldsymbol{\kappa}} \leftarrow \hat{\boldsymbol{\kappa}} + \hat{\boldsymbol{\kappa}}^{(m)}$
  $\hat{\boldsymbol{\nu}} \leftarrow \hat{\boldsymbol{\nu}} + \hat{\boldsymbol{\nu}}^{(m)}$
end for
for $j = 1 : k$ **do**
  $\hat{z}_{ij} \leftarrow$ Use equation (3.9), with $\hat{\boldsymbol{p}}/\text{MCIters}$, $\hat{\boldsymbol{\mu}}/\text{MCIters}$, $\hat{\boldsymbol{\Sigma}}/\text{MCIters}$ and each data point (i-th) in S_n
end for
$z_i \leftarrow \text{argmax}_{j \in \{1,\ldots,k\}} \hat{z}_{ij}$ For each data point (i-th) in S_n
$\hat{z}^{'} \leftarrow \text{argmax}_{j \in \{1,\ldots,k\}} \hat{z}_j^{'}$
return $\hat{y}/\text{MCIters}$, $\hat{z}^{'}$, $\mathbf{z}$, $\hat{\boldsymbol{\Sigma}}/\text{MCIters}$, $\hat{\boldsymbol{\mu}}/\text{MCIters}$, $\hat{\boldsymbol{p}}/\text{MCIters}$, $\hat{\boldsymbol{\Psi}}/\text{MCIters}$, $\hat{\boldsymbol{m}}/\text{MCIters}$, $\hat{\boldsymbol{\kappa}}/\text{MCIters}, \hat{\boldsymbol{\nu}}/\text{MCIters}$

number of levels $L_{\max}$, maximum number of parents $n_{\text{Pa,max}}$, probability of a parent to be an activator $p_{\text{activator}}$, maximum possible mutation probability $\text{mut.}_{\max}$, minimum possible

Table 3.2: Input Parameters Used in Generating Pathways.

(number of nodes) $n_{\text{nodes}} = 8$	(minimum number of mutations) $n_{\text{mut.,min}} = 3$
(maximum level) $L_{\text{max}} = 3$	(maximum number of mutations) $n_{\text{mut.,max}} = 7$
(minimum level) $L_{\text{min}} = 3$	(minimum mutation probability) $\text{mut.}_{min} = 5\%$
(number of first-level genes) $m = 3$	(maximum mutation probability) $\text{mut.}_{\text{max}} = 25\%$
(maximum number of parents) $n_{\text{Pa,max}} \sim U(3, 6)$	(probability of a deletion type mutation) $p_{\text{mut.type}} = 0.5$
(probability of an edge to be APS) $p_{\text{activator}} = 0.5$	

mutation probability mut._{min}, maximum possible number of mutations $n_{\text{mut.,max}}$, minimum possible number of mutations $n_{\text{mut.,min}}$, probability of a mutation to be deletion of an edge $p_{\text{mut.type}}$, number of the first-level genes m. The specific values selected for synthesizing the pathways in our simulations are provided in Table 3.2.

To begin, the first component's pathway is synthesized as the original network. Then the pathways of other components are generated by perturbing the original network via mutations, which include deletion of an edge or changes in regulation types.

The first component's pathway is generated based on the following procedure:

- Comp. 1:

  1. Number of levels, $L \sim U(L_{\text{min}}, L_{\text{max}})$.
  2. For a fixed n and L, place two nodes at the first level and one node at all other levels.
  3. Randomly select all the other remaining nodes' levels from $U(1, L)$.
  4. For a given node:

     -(candidate parents) $\text{pa.}_{\text{candid.}}$: all the nodes in higher levels than the child node itself.

     -(number of parents) $n_{\text{pa.}} \sim U(1, min(n_{\text{Pa,max}}, |\text{pa.}_{\text{candid.}}|))$.

     -Determine segment type: $Ber(p_{\text{activator}})$.

Other components' pathways are generated based on the following procedure:

- Comp. k ($k \geq 2$): In order to generate the pathways associated with the kth component, we

randomly *mutate* the edges (regulations) as follows:

$$n^k_{\text{mut.}} \sim U(\text{mut.}^k_{\text{min}}, \text{mut.}^k_{\text{max}}),$$
$$\text{mut.}^k_{\text{max}} = min\{\lfloor(\text{mut.}_{max} \times |\mathcal{R}|)\rfloor, n_{\text{mut.},max}\},$$
$$\text{mut.}^k_{\text{min}} = max\{\lfloor(\text{mut.}_{min} \times |\mathcal{R}|)\rfloor, n_{\text{mut.},min}\}),$$

where $|\mathcal{R}|$ is the number of all regulations in the original (first component's) pathway and $n^k_{\text{mut.}}$ is the number of mutations drawn for component k. The edges to be affected by mutation are randomly selected from the set of all the edges of the original graph (first component's pathway) and the type of mutation (deletion of an edge or change in regulation type) for each selected edge is randomly picked from $Ber(p_{\text{mut.type}})$.

3.3.1.2 Generating data from the synthetic pathways

The index of the target gene is randomly picked from $U(n-3, n)$. To generate data from the pathway structure, fix $\mu_{\text{min}} = 1.5$, $\mu_{\text{max}} = 3.5$, $\sigma^2 = 1$, $\rho_{\text{min}} = 0.15$, $\rho_{\text{max}} = 0.35$, and $\sigma_n^2 = 0.05$, and then do the following:

- Comp. 1: The mean and covariance matrix of the genes (nodes) of the first level are fixed (m is the number of genes in the first level), where the mean vector is $\boldsymbol{\mu}_0 = (\mu_{x_1}, ..., \mu_{x_m})$ and $\mu_{x_1} = ... = \mu_{x_m} \sim U(\mu_{\text{min}}, \mu_{\text{max}})$, and the covariance matrix is

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} \sigma^2 & \rho\sigma^2 & \dots & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 & \dots & \rho\sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ \rho\sigma^2 & \dots & \dots & \sigma^2 \end{bmatrix}_{m\times m},$$

with $\rho \sim U(\rho_{\text{min}}, \rho_{\text{max}})$. All other remaining genes are assumed to follow the following linear dependency [39]. For each gene i, $x_i = a_i^T \boldsymbol{x}_{\boldsymbol{pa_i}} + \mathcal{N}(0, \sigma_n^2)$, by which the Gaussian assumption is kept. In fact, this linear relationship determines the marginal distribution of

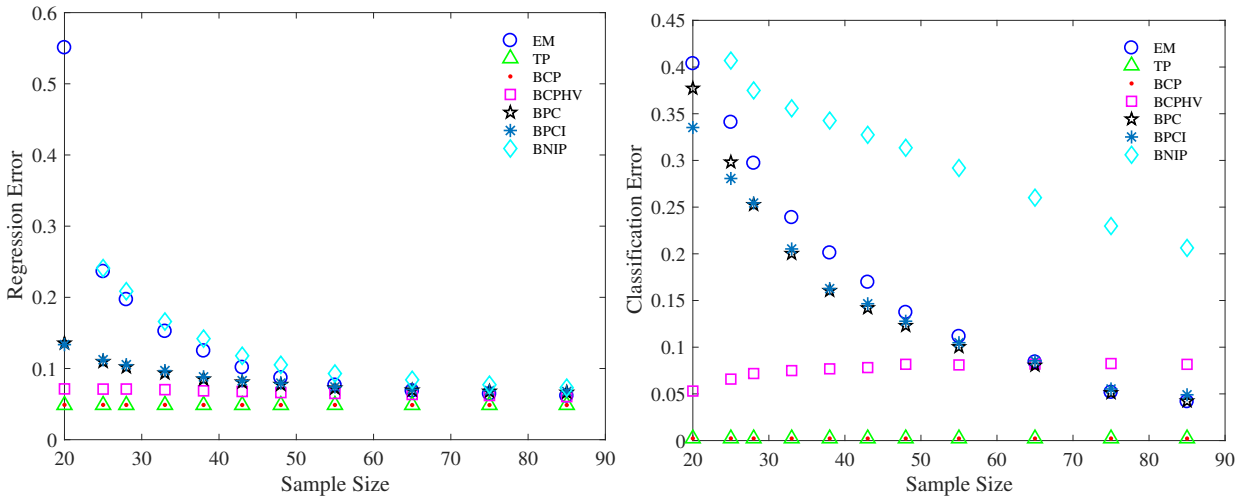
gene i, and also the joint distribution of all genes that are all Gaussian. Here, $\boldsymbol{x_{pa_i}}$ and a_i represent the set of all parents of node i and their corresponding coefficients, respectively. The coefficients are set as $|a_i(j)| = \frac{1}{N_i}$ for $j = 1, ..., N_i$, where N_i is the number of parents of node i, and their signs are determined by the type of influence of the parent node, positive for activation and negative for repression. By this linear relationship, the marginal distribution of gene i is $P(x_i) = \mathcal{N}(a_i^T \boldsymbol{\mu}_{\boldsymbol{x_{pa_i}}}, a_i \boldsymbol{\Sigma}_{\boldsymbol{x_{pa_i}}} a_i^T + \sigma_n^2)$, where $\boldsymbol{\mu}_{\boldsymbol{x_{pa_i}}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x_{pa_i}}}$ denote the mean vector and covariance matrix of the parents of gene i, respectively.

- Comp. 2: Similar to the above setup with $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_1 = 1.5\boldsymbol{\Sigma}_0$.

3.3.1.3 Results

In this section we compare seven different methods relative to classification and regression errors. Since expectation maximization (EM) is the most practical alternative, the major comparison is between EM, the proposed Bayesian prior construction with one iteration of the prior construction and update method (BPC), and multiple iterations of the proposed prior construction and update method (BPCI). We shall also consider Bayesian with (data dependent) non-informative prior (BNIP) [78, 79]. For illustration, we will consider Bayesian with a prior centered on the true parameter values and having low variance, meaning large κ_i and ν_i (BCP); Bayesian with a prior centered on the true parameter values and having high variance, meaning small κ_i and ν_i (BCPHV); and simply plugging in the true parameters (TP). In real-world applications we lack knowledge of the true parameter values; however, TP, BCP and BCPHV provide comparisons to show how well the other practical alternatives are performing. For GMM, improper priors result in improper posteriors and cannot be used [79]. Furthermore, Bayesian GMM inference suffers from several issues, including label switching [80, 81]. Therefore, for comparison with a relatively non-informative prior, we have followed the approach in [78, 79] and assumed some true identifiability constraints on the mixture probabilities (an ordering of mixture probabilities). To have a fair comparison, the initial labels for the non-informative case's chain are also calculated by EM. We have observed that by constructing the GMM prior, the label switching problem in the MCMC

chain ceases to exist, the reason being discriminative priors. We have simulated 200 pairs of random pathways. Two different setups are considered for the mixing probabilities. In one the mixing probabilities (p_1 and p_2) are set to 0.6 and 0.4 for the first and second components, respectively, and in the other one these are set to 0.72 and 0.28.

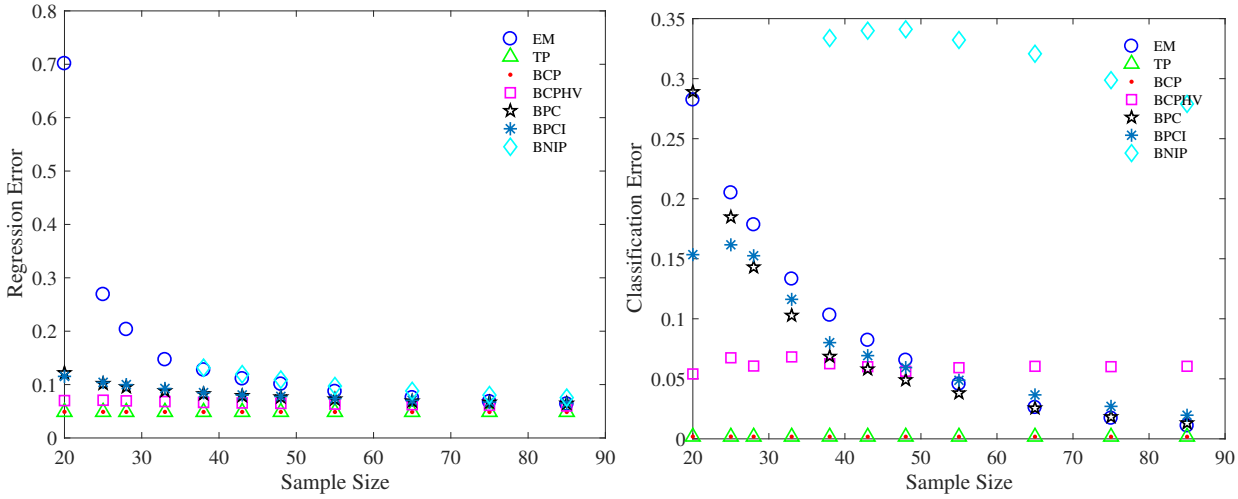


(a) Regression (synth. path.)

(b) Classification (synth. path.)

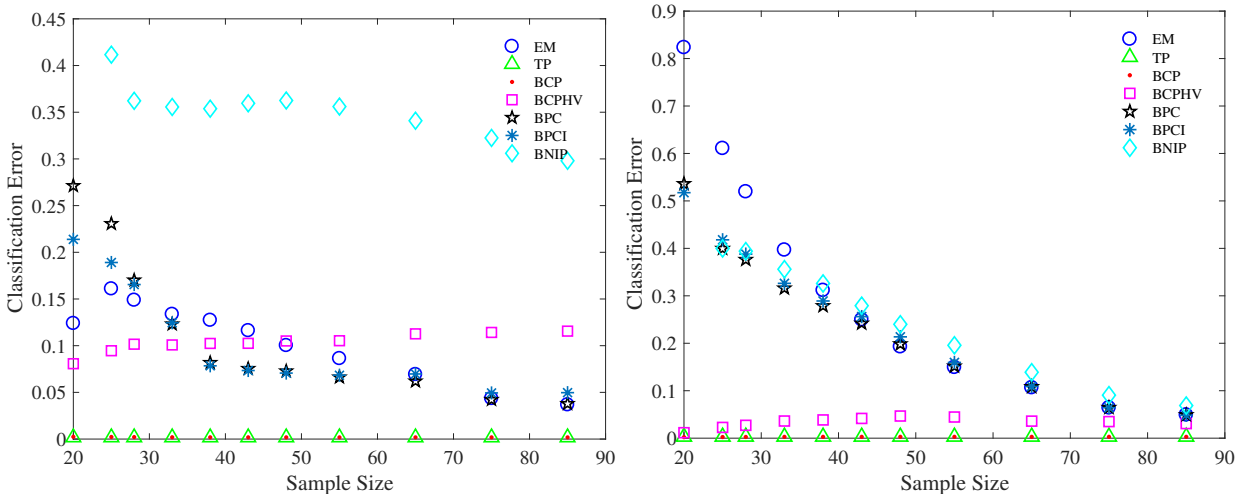
Figure 3.3: Average regression and classification errors on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$ in the top and bottom panels respectively.

For each pair of pathways, the simulations are performed with different sample sizes. For a fixed pair of pathways, and a fixed sample size, there are 40 repetitions of training and test data generation. For regression errors, fixing the pathways, sample size and repetition, the average regression error (mean-square error) on 1,000 test samples is calculated. For classification errors, in each run, fixing the pathways, sample size and repetition, 1,000 complete test data points are classified based on the GMM model each time by (i) plugging-in the inferred parameter values (estimates of parameters) and using (3.17) for EM and TP, or (ii) by performing OBC for Bayesian methods (BPC, BPCI, BNIP, BCP, BCPHV). The classification error ($\hat{E}rr$) on these test points is calculated based on $\hat{E}rr = p_1\hat{E}rr_1 + p_2\hat{E}rr_2$, where $\hat{E}rr_1$ and $\hat{E}rr_2$ are the component-conditional classification errors, i.e. these are the mean classification errors on the test data points belonging



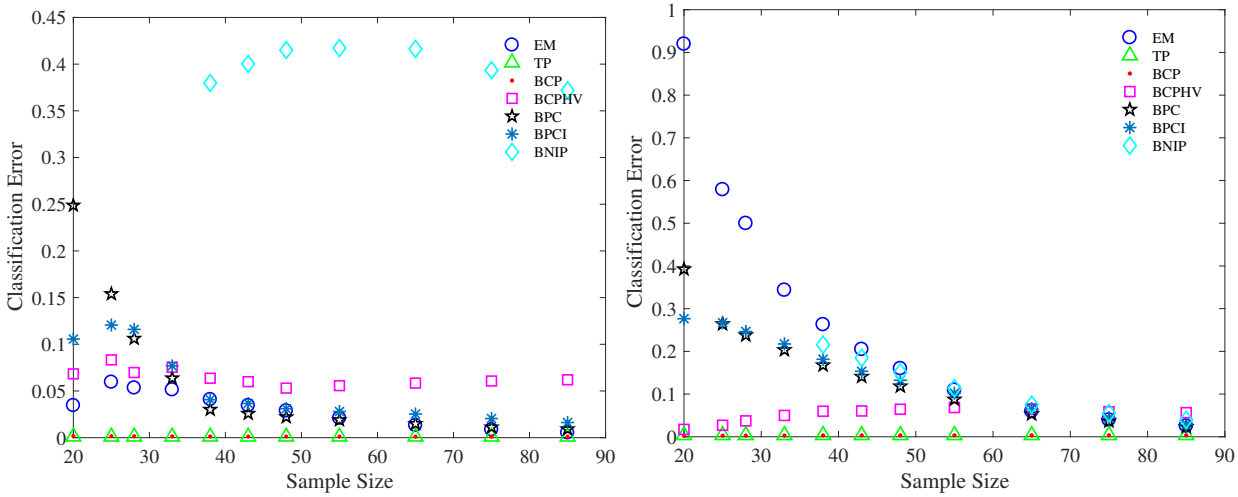
(a) Regression (synth. path.) (b) Classification (synth. path.)

Figure 3.4: Average regression and classification errors on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$ in the top and bottom panels respectively.



(a) Component 1 (synth. path.) (b) Component 2 (synth. path.)

Figure 3.5: Average component-conditional classification errors on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$ for the first and second components in the top and bottom panels respectively.



(a) Component 1 (synth. path.)

(b) Component 2 (synth. path.)

Figure 3.6: Average component-conditional classification errors on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$ for the first and second components in the top and bottom panels respectively.

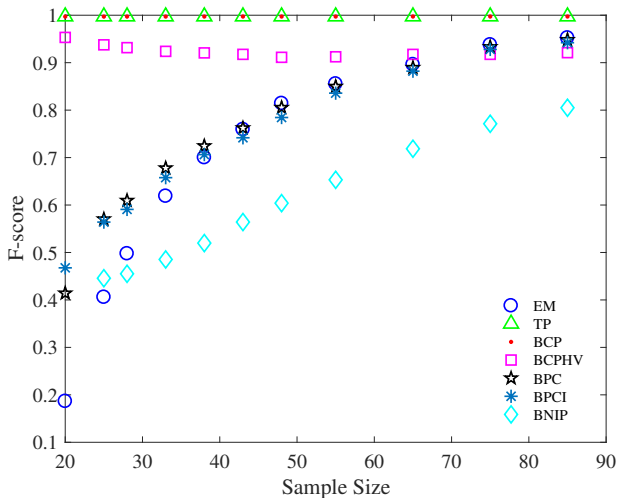


Figure 3.7: Average F-score on synthetic pathways with $p_1 = 0.6$ and $p_2 = 0.4$.

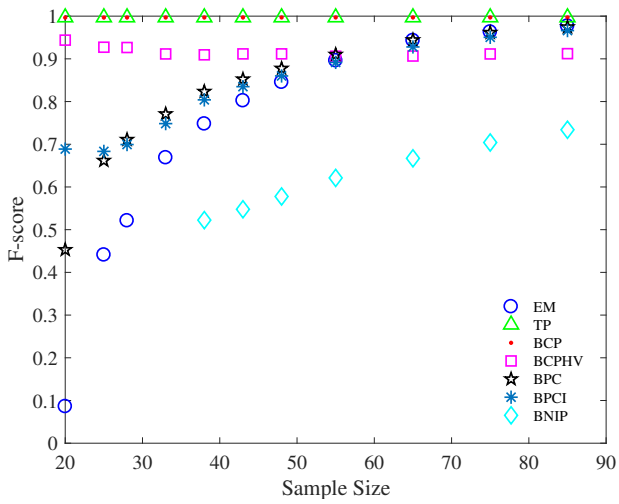
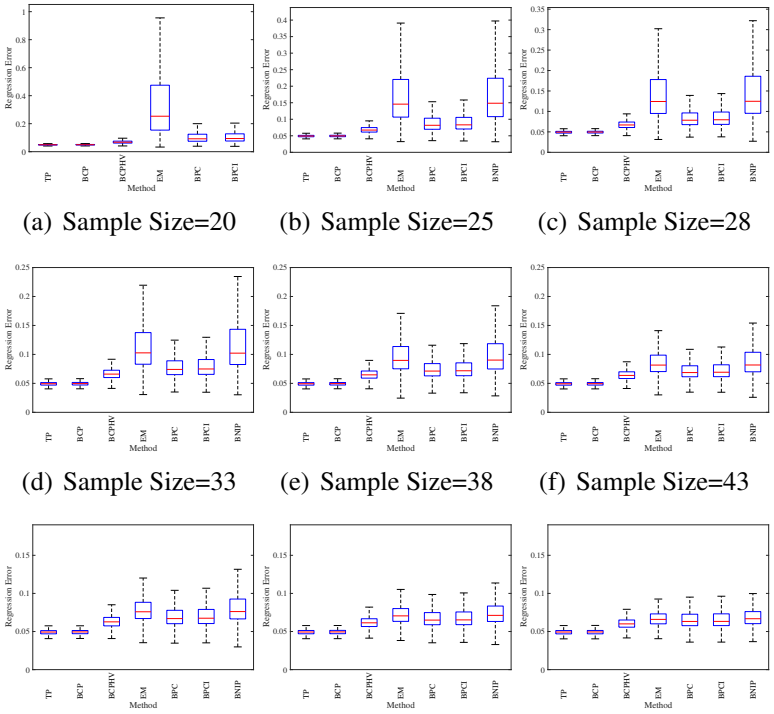


Figure 3.8: Average F-score on synthetic pathways with $p_1 = 0.72$ and $p_2 = 0.28$.

to the first and second components, respectively.

The average regression and classification errors over all the networks and repetitions are shown as functions of sample size for mixing probabilities of 0.6 and 0.4 in Fig. 3.3(a) and Fig. 3.3(b), and for mixing probabilities of 0.72 and 0.28 in Fig. 3.4(a) and Fig. 3.4(b), respectively. Note that for BNIP, a sufficient number of data points is required to get a proper posterior, so that the error line for this method starts from the sample size that results in proper posteriors. The average component-conditional classification errors over all the networks and repetitions for both of the components are depicted vs the sample size for the mixing probabilities of 0.6 and 0.4 in Fig. 3.5(a) and Fig. 3.5(b), and for mixing probabilities of 0.72 and 0.28 in Fig. 3.6(a) and Fig. 3.6(b). Moreover, the average F-score (geometric mean of precision and recall) over all the networks and repetitions is shown as function of sample size for mixing probabilities of 0.6 and 0.4 and mixing probabilities of 0.72 and 0.28 in Fig. 3.7 and Fig. 3.8, respectively. Box plots of the regression and classification errors over all the networks and all the repetitions for mixing probabilities of 0.6 and 0.4 are shown for different sample sizes in Fig. 3.9 and Fig. 3.10, respectively. The corresponding figures for the mixing probabilities of 0.72 and 0.28 are included in Appendix A.1 in the supplementary materials.

Figure 3.3(a) and Fig. 3.4(a) show that, for regression, the multiple iterations of BPCI have

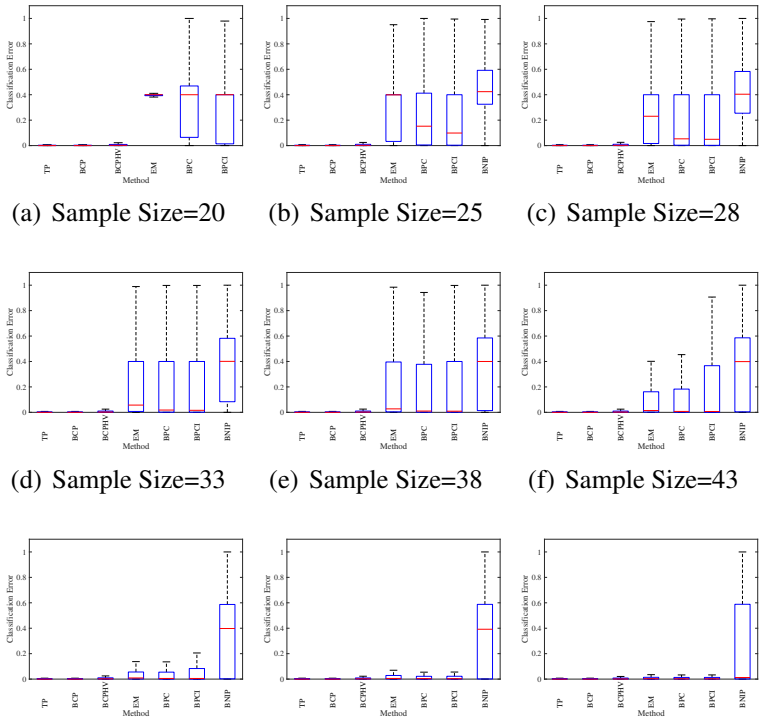


(a) Sample Size=20 (b) Sample Size=25 (c) Sample Size=28

(d) Sample Size=33 (e) Sample Size=38 (f) Sample Size=43

(g) Sample Size=48 (h) Sample Size=55 (i) Sample Size=65

Figure 3.9: Box plots of regression errors on synthetic pathways for different sample sizes with $p_1 = 0.6$ and $p_2 = 0.4$.



(a) Sample Size=20 (b) Sample Size=25 (c) Sample Size=28

(d) Sample Size=33 (e) Sample Size=38 (f) Sample Size=43

(g) Sample Size=48 (h) Sample Size=55 (i) Sample Size=65

Figure 3.10: Box plots of classification errors on synthetic pathways for different sample sizes with $p_1 = 0.6$ and $p_2 = 0.4$.

very little advantage over the single iteration of BPC; and both significantly outperform EM for very small samples and maintain some advantage up to about 65 data points, which is more than what is available in many studies. On the other hand, EM always outperforms Bayesian with a non-informative prior (BNIP). Regarding the ideal methods, TP must be the best and a tight correctly centered prior (BCP) performs virtually the same (for very small sample size TP has a tiny advantage over BCP but this is not visible in the graph). As expected, a correctly centered prior with larger variance (BCPHV) performs worse than BCP but slowly gains ground as the sample size increases. From our perspective, what is important is that, even with very small sample sizes, both BPC and BPCI perform close to BCPHV.

Regarding Fig. 3.3(b) and Fig. 3.4(b), similar comments apply to EM, BPC, BPCI, and BNIP, except that the advantage of BPC and BPCI over EM is not so great for small samples; nevertheless, the proposed Bayesian prior construction approach still outperforms EM for the cases with sample sizes up to about 65 data points. Also, the advantage of BPCI over BPC for very small sample sizes is more clear here, though this advantage vanishes as the sample size increases. Moreover, as the sample size increases, BPC and BPCI outperform BCPHV. Figure 3.7 and Fig. 3.8 further confirm the advantage of BPC and BPCI over EM for small sample sizes. Also, it can be seen that for very small sample sizes BPCI performs better than BPC based on the F-score metric.

Comparing Fig. 3.3(a) and Fig. 3.3(b), and also Fig. 3.4(a) and Fig. 3.4(b), we notice that the behavior of BCPHV differs for regression and classification. For regression, the error monotonically decreases with increasing sample size. But for classification, the error first grows and then decreases (the decrease not being seen in the figure because the tested sample size stops at 85). Although this behavior is not germane to GMM prior construction, we would like to conjecture as to what is happening. In the MCMC chain, z_{ij} are sampled from a multinomial distribution. Since sampling is random, there is shrinkage of the distance between parameters of different Gaussians. For example, x_i may belong to component 1; but in the MCMC chain z_i's outcome corresponding to x_i sampled in each chain iteration might sometimes be component 2 instead of component 1. This can cause the component's parameters sampled in the same iteration to get a little bit closer

to each other. The test data's weights (probability of belonging to a specific component) are calculated in each chain iteration based on the sampled parameters. For regression, since it is a weighted average in each iteration, slight changes of these weights have negligible effect. However, for classification, since the component (class) with the highest sum of the weights calculated during the chain for the test data is chosen as the classification output, the performance is more affected by this phenomena. When the sample size is large, estimations become accurate and, as typical with Bayesian estimation, asymptotic behavior of the sampling becomes prominent.

To the best of our knowledge there is no other existing method that can incorporate information in the form of signaling pathways with regulatory relationships along with unlabeled data (in a mixture setup) for regression and classification purposes. Nevertheless, comparison results of our proposed method with the method of [82] (hereafter and in the Appendix referred to as GRACE) that uses the connectivity information in the pathways for regression problems, but not the regulating information, for a single component regression problem are provided in Appendix A.2. The pathway and data generation setup used for that comparison is the same as the procedure described in this section, except that only one component is used for (training and test) data generation. In the single-component regression-problem comparison based on synthetic pathways and data, our method outperforms the method of [82]. More details are provided in Appendix A.2.

3.3.2 Performance on a Colon Cancer Pathway

In this section the performance of the seven methods are evaluated on the (synthetic) data generated based on the colon caner pathways in Fig. 3.11 [39].

We followed the approach in [71, 39] by employing a simplified model from three basic pathways: Ras/Raf/Mek, PI3K, and JAK/STAT, which can model the genome behavior of colon cancer [71, 39]. The interactions are shown in Fig. 3.11. We assume that the samples are unlabeled, with samples from both tumor and normal cases (where the classification is between normal and tumor/cancer cases). Since MEK1/2 is a common downstream marker for colon cancer, the target for regression analysis is considered to be MEK1/2, i.e. the regression task is predicting the expression of MEK1/2. We assume that for the cancer samples, TSC1/TSC2 is stuck at zero [39],

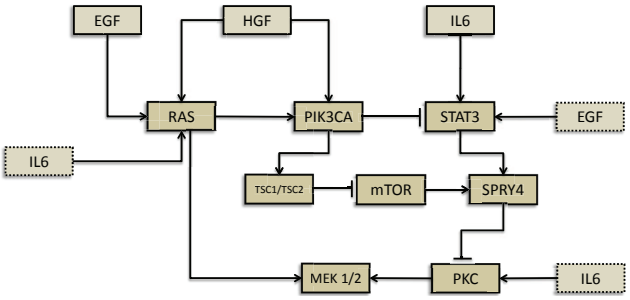
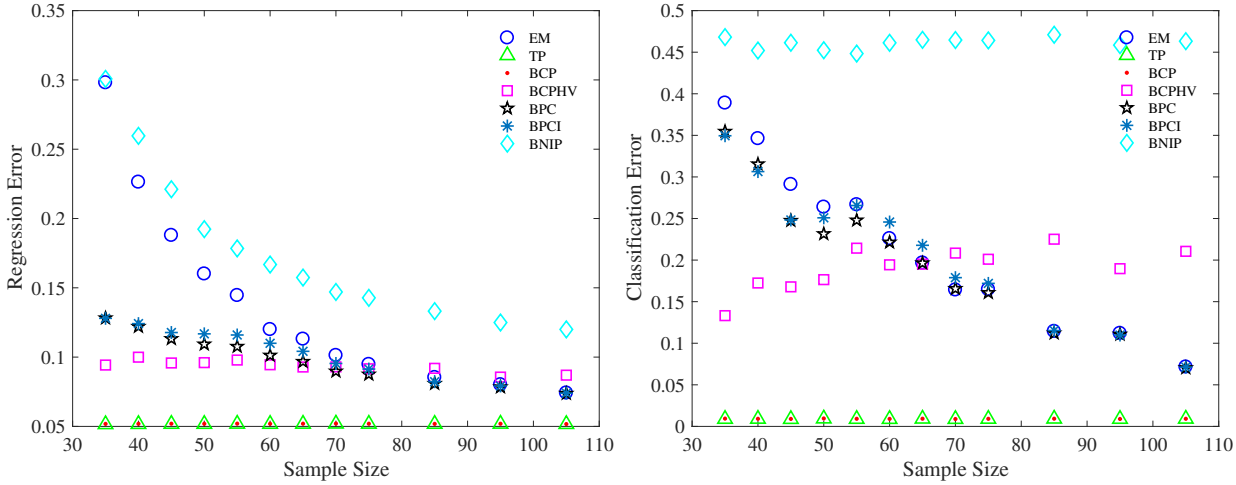
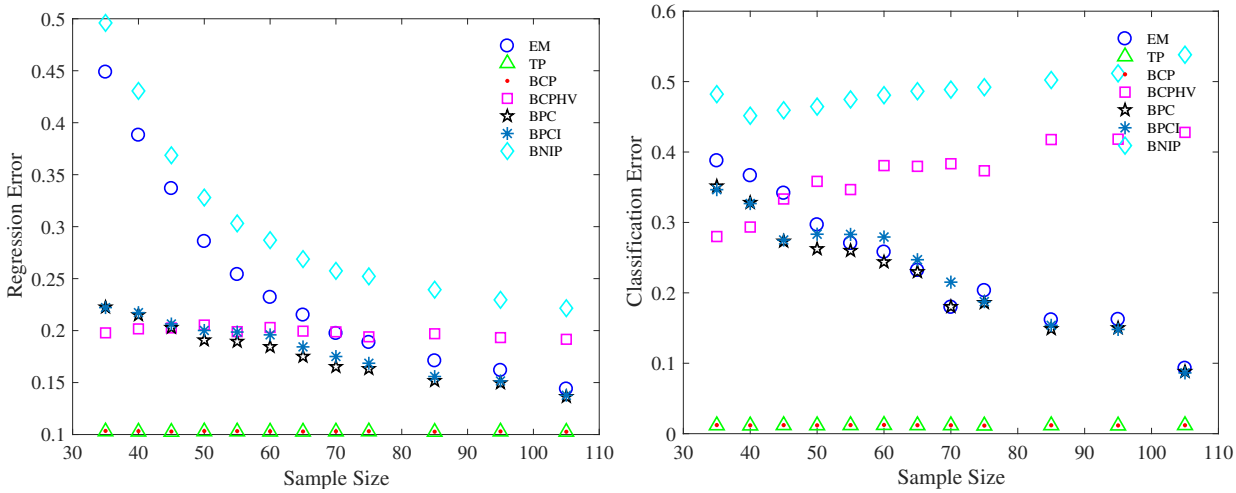


Figure 3.11: A simplified colon-cancer-related pathway. Reprinted from [39, 88].



(a) Regression (colon. path.) (b) Classification (colon. path.)

Figure 3.12: Performance on colon cancer pathways in Fig. 3.11. Average regression and classification errors with $\sigma_n^2 = 0.05$ in the top and bottom panels respectively.



(a) Regression (colon. path.) (b) Classification (colon. path.)

Figure 3.13: Performance on colon cancer pathways in Fig. 3.11. Average regression and classification errors with $\sigma_n^2 = 0.1$ in the top and bottom panels respectively.

and also the regulation type of Ras on MEK1/2 is changed. Data generation from the pathways is similar to Section 3.3.1.2 with $p_1 = 0.6$ and $p_2 = 0.4$, except that here the means of the upstream genes EGF, HGF, and IL6 are all set to 1.5, $\rho = 0.2$, $\sigma^2 = 1$, and two levels of noise $\sigma_n^2 = 0.05$ and $\sigma_n^2 = 0.1$ are considered. Here, the first and second components correspond to normal and tumor/cancerous cases, respectively.

The linear relationships for the first component (normal case) for the downstream genes are given by

$$\begin{aligned}
\text{Ras} &= \frac{1}{3}\text{EGF} + \frac{1}{3}\text{HGF} + \frac{1}{3}\text{IL6} + \epsilon; \\
\text{PIK3CA} &= \frac{1}{2}\text{HGF} + \frac{1}{2}\text{Ras} + \epsilon; \\
\text{STAT3} &= \frac{1}{3}\text{EGF} + \frac{1}{3}\text{IL6} + \frac{1}{3}\text{PIK3CA} + \epsilon; \\
\text{TSC1/TSC2} &= \text{PIK3CA} + \epsilon; \\
\text{mTORC1} &= -\text{TSC1/TSC2} + \epsilon; \\
\text{SPYR4} &= \frac{1}{2}\text{STAT3} + \frac{1}{2}\text{mTORC1} + \epsilon; \\
\text{PKC} &= \frac{1}{2}\text{IL6} - \frac{1}{2}\text{SPYR4} + \epsilon; \\
\text{MEK1/2} &= \frac{1}{2}\text{Ras} + \frac{1}{2}\text{PKC} + \epsilon,
\end{aligned}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. For the second component (cancer), these equations hold except for MEK1/2 and TSC1/TSC2, which become

$$\begin{aligned}
\text{TSC1/TSC2} &= \epsilon; \\
\text{MEK1/2} &= -\frac{1}{2}\text{Ras} + \frac{1}{2}\text{PKC} + \epsilon.
\end{aligned}$$

For a fixed sample size, 300 sets of training and test data (1000 test data points) are generated and the average regression error (mean-square error) and average classification error based on $\hat{Err} = p_1\hat{Err}_1 + p_2\hat{Err}_2$, where $\hat{Err}_1$ and $\hat{Err}_2$ are the component-conditional classification

errors, is calculated. The results of mean regression and classification errors with respect to different sample sizes for $\sigma_n^2 = 0.05$ are shown in Fig. 3.12(a) and Fig. 3.12(b), respectively. Similar results of mean regression and classification errors for $\sigma_n^2 = 0.1$ are depicted in Fig. 3.13(a) and Fig. 3.13(b), respectively. Box plots of regression and classification errors over all repetitions for different sample sizes, average component-conditional classification errors over all the repetitions for both of the components, and average F-score over all the repetitions as a function of sample size for both of the noise levels are provided in Appendix A.3. It can be seen from Fig. 3.12(a)-Fig. 3.13(b) that BPC and BPCI outperform EM and BNIP in both regression and classification for small sample sizes (up to about 85 data points for regression and 65 data points for classification). As the sample size increases, BPC and BPCI outperform BCPHV. These comparison results on (simulated) data generated based on a set of real pathways further confirm the advantage of BPC and BPCI over EM for small sample sizes.

4. OPTIMAL BAYESIAN SUPERVISED DOMAIN ADAPTATION FOR RNA SEQUENCING DATA *

4.1 Introduction

In this Chapter, we aim to develop a framework to leverage data from other *domains* to design better predictors in the *target* domain of interest in addition to benefiting from the available *a priori* information. When designing predictive models for a target task, traditionally only the data from the target domain are used for training with the commonly adopted assumption that the training and testing data have the same feature-label distributions. However, in many cases, especially with next-generation sequencing (NGS) technologies, the number of training samples that can be collected in the target domain is limited compared with the dimensionality of the features (the number of genes). Collecting appropriate data from complex diseases is a costly procedure, if not prohibitive, considering the clinical, biological, and technical challenges involved in the process. These limitations can prohibit collecting enough samples from the disease/condition of interest to design a reproducible predictor. Given the prevalent data heterogeneity in complex diseases like cancer [4], usually more samples are needed than what can be collected to achieve reliable predictors. It is believed that different diseases share some underlying biological processes and modules [83, 84, 85, 86], indicating that data from one disease can be informative for other diseases. Hence, it is desirable to learn useful information from available data from other conditions and/or technologies to help derive more accurate predictions in the target domain. Moreover, other than the data at hand, additional knowledge is usually available *a priori* (before observing data) that can be beneficial for the target task [87, 88, 89], as also seen in Chapters 2 and 3. Examples of this include interaction networks, which might have been compiled from several studies and databases [85, 90, 91] containing potentially useful information for the target task. Our goal is to develop a new optimal Bayesian supervised domain adaptation (OBSDA) framework capable of leveraging

*Reprinted with permission from S. Boluki, X. Qian, and E. R. Dougherty, “Optimal Bayesian supervised domain adaptation for RNA sequencing data," Bioinformatics, 2021, 10.1093/bioinformatics/btab228. Copyright 2021 OUP.

data and label information from other domains in addition to prior network knowledge to design more accurate and reliable predictors in a target domain of interest.

Transfer learning and domain adaptation methods [92, 93] aim to leverage data from other domains for achieving better results for the task in the target domain. Common approaches generally include adapting the predictor in the source domain to the target domain and/or the distribution of the data across domains [94]. Some methods, including [95, 96] reweight the source and target samples. Other representative methods, such as [97, 98], first project the target and all or a subset of source data to a common subspace, which minimizes a discrepancy metric between the marginal distributions of features in the domains, and then train a discriminator in that space. The application of these methods are often limited to cases where source and target data are from the same classes. On the other hand, multi-task learning methods [99, 100, 101] aim to improve prediction power overall in all domains/tasks, with some requiring at least several tasks/domains for reasonable performance. The majority of deep learning-based domain adaptation methods [102, 103, 104], which usually share parameters and/or lower-level representations across domains and have found their major successes in computer vision tasks, need much larger training sets in all the domains than what is practical in typical clinical studies.

Some of the recent transfer learning and domain adaptation works on gene expression data include [105, 106, 107, 108]. In [105] the authors developed a method to predict differentially expressed genes in a condition for humans based on gene expression data collected from disease studies on mice. [106] proposed two methods respectively—mapping of features to a common subspace and mapping target domains to the source space—to better predict drug sensitivity based on gene expression data from additional databases. Both [107] and [108] proposed methods for utilizing gene expression data from other domains to build more reliable cancer subtype predictors in the target domain. In [107], a hierarchical Bayesian model was developed to map the samples from different domains to a shared latent space with the classifier trained on the lower dimensional representations to predict cancer subtypes. One shortcoming of the method is that label information is not used in the latent representation learning stage. [108] proposed a Bayesian method

with joint priors on the parameters from source and target domains and derived the predictor by marginalizing over source parameters. Despite being a principled approach, it models only the relationship between data from the same classes across domains, with the limitation of not fully benefiting from the available data. More critically, neither of these methods can use additional interaction network knowledge as prior biological knowledge in their framework.

We propose a new Bayesian framework for supervised domain adaptation for NGS count data, with generative models utilizing both data and label information from multiple domains to learn shared genes embedding and domain and label-dependent latent parameters. Through a hierarchical Bayesian structure and a factorization setup of parameters with a subset of global random variables, useful information from all the domains and labels can be leveraged for cancer subtype prediction in the target domain. The domains can include data from the same labels as or different labels than the target domain. We use negative binomial likelihoods to model RNA-Seq count data considering potential sample heterogeneity to obviate the need for *ad-hoc* preprocessing steps. The predictor in our method is based on the concept of optimal Bayesian operator design [109], where the predictor is derived point-wise by comparing the posterior expectation of the class-conditional likelihoods for a given sample. Moreover, our framework can take advantage of the available prior knowledge in terms of gene-gene interaction networks to derive more accurate and generalizable predictors in the target domain.

In the following sections, we first introduce our basic OBSDA model and derive an efficient Gibbs sampler by exploiting novel data augmentation techniques for the negative binomial distribution [110]. Then, we propose an extension of OBSDA with flexible semi-implicit variational inference [111]—SI-OBSDA—that employs explicit distributions mixed with implicit neural network generators. We then show how we can incorporate prior interaction network knowledge in SI-OBSDA for informed inference. Finally, we verify the benefits of our OBSDA and SI-OBSDA by providing results for comparing our methods with single-domain and multi-domain baselines on predicting cancer subtypes with The Cancer Genome Atlas (TCGA) RNA-Seq data.

4.2 Methods

4.2.1 OBSDA

The negative binomial (NB) distribution is a popular choice to model overdispersion in RNA-Seq count data due to technical and biological variations [112, 113]. Let $\mathbf{x} \sim \text{NB}(r, p)$, which is a NB distribution with the probability mass function (PMF) $\frac{\Gamma(\mathbf{x}+r)}{\mathbf{x}!\Gamma(r)}(p)^{\mathbf{x}}(1-p)^r$ with the count data $\mathbf{x} \in \{0, 1, 2, \cdots\}$ and $\Gamma(\cdot)$ being the gamma function. Denoting the observed count for gene j in sample i of domain d with label l by $\mathbf{x}^l_{d,j,i}$, and the collection of all genes for that sample by $\mathbf{x}^l_{d,i}$, we model the counts from multiple domains (sources) by a factorization of the parameters as

$$\mathbf{x}^l_{d,i} \sim \text{NB}(\boldsymbol{\Phi}\boldsymbol{\theta}^l_d, p^l_{d,i}). \tag{4.1}$$

Here, $\boldsymbol{\Phi} \in \mathbb{R}^+_{J\times K}$, with rows $\boldsymbol{\phi}^T_j \in \mathbb{R}^+_{1\times K}$ for $j = \{1, \cdots, J\}$, is the matrix quantifying the association between the genes and latent factors. This association is gene dependent, but for each domain and label the relevancy of the factors is different. The relevancy of the factors to each domain and label is quantified by $\boldsymbol{\theta}^l_d$. We model each element of $\boldsymbol{\theta}^l_d$ with a Gamma distribution, $\boldsymbol{\theta}^l_{d,k} \sim \text{Gamma}(u_{d,k}, \frac{1}{v^l})$, where v^l is label dependent and $u_{d,k}$ is domain dependent. In other words, the domain and label dependencies are decomposed into the two sets of parameters to help identifiability and share signals across domains and labels. The Gamma distribution encourages sparsity in the model, where each class in each domain can select a few of latent factors as relevant. We place the Gamma prior on the label-dependent parameters v^l. To enable domain-dependent latent representations, we assume $u_{d,k} \sim \text{Gamma}(b_k, \frac{1}{q_d})$, where b_k and q_d represent the global latent factor and domain-specific parameters. $p^l_{d,i}$ accounts for the potential sample heterogeneity in a class of a domain.

Note that unlike factor analysis models [114, 115, 107] where the observations are factorized, here a latent variable of the model is factorized, and is learned jointly with other latent variables in the model using the data from multiple domains. Moreover, we leverage the label information in a supervised setting in contrast with standard factor analysis.

As a factorization model, $\mathbf{x}^l_{d,j,i} \sim \text{NB}(\boldsymbol{\phi}_j^T \boldsymbol{\theta}^l_d, p^l_{d,i})$ can be augmented as $\mathbf{x}^l_{d,j,i} = \sum_{k=1}^K \mathbf{x}^l_{d,j,i,k}$, where $\mathbf{x}^l_{d,j,i,k} \sim \text{NB}(\boldsymbol{\phi}_{j,k}\boldsymbol{\theta}^l_{d,k}, p^l_{d,i})$, and the expected expression of gene j in sample i of domain d with class label l can be expressed as

$$\mathbb{E}[\mathbf{x}^l_{d,j,i}] = \big(\sum_{k=1}^K \boldsymbol{\phi}_{j,k}\boldsymbol{\theta}^l_d\big)\frac{p^l_{d,i}}{1-p^l_{d,i}}. \tag{4.2}$$

The expectation can be interpreted as the true abundance of a gene adjusted by potential data heterogeneity in a class of a domain, removing the need for *ad-hoc* normalization steps. More specifically, the true abundance is comprised of the contributions of all latent factors, where each contribution is encoded as the product of the association between a gene and a factor and the relevancy of that factor for the domain and class.

The factors can be seen as underlying biological processes or functional modules relating to or causing genotypic or phenotypic changes. K is the number of such factors considered in the model and is a hyperparameter. From the modeling perspective, the random variables corresponding to the association between the genes and the underlying biological processes (factors) are assumed to be the same across domains and labels. In other words, the contribution of each underlying biological process to the expression of a gene depends on both the gene and process relationship, which is encoded by a global variable and shared across domains and labels, and the relevancy of the process to the specific label/class in the domain, which is domain and label dependent and learned from data.

It is worth noting that the OBSDA model can be seen as sharing knowledge across the different labels in the same domain as well as across domains for more robust estimations. Moreover, it can integrate data from domains containing different labels, i.e. where a one-to-one correspondence between labels across domains does not exist. These properties will especially be helpful when the number of samples is low in the target domain.

We complete the model by placing conjugate priors for the rest of the parameters as follows:

$$
\begin{aligned}
&\mathbf{x}^l_{d,j,i} \sim \text{NB}(\boldsymbol{\phi}_j^T \boldsymbol{\theta}^l_d, p^l_{d,i}) \\
&\boldsymbol{\theta}^l_{d,k} \sim \text{Gamma}(u_{d,k}, \frac{1}{v^l}), \quad u_{d,k} \sim \text{Gamma}(b_k, \frac{1}{q_d}) \\
&v^l \sim \text{Gamma}(e_0, \frac{1}{f_0}), \quad b_k \sim \text{Gamma}(\frac{\gamma_0}{K}, \frac{1}{c_0}) \\
&q_d \sim \text{Gamma}(w_0, \frac{1}{u_0}), \quad (\phi_{1,k}, \cdots, \phi_{J,k}) \sim \text{Dir}(\eta, \cdots, \eta) \\
&p^l_{d,i} \sim \text{Beta}(g_0, h_0), \quad c_0 \sim \text{Gamma}(a_0, \frac{1}{d_0}) \\
&\gamma_0 \sim \text{Gamma}(\alpha_0, \frac{1}{\beta_0}),
\end{aligned}
\tag{4.3}
$$

where we have exploited the beta-negative binomial, gamma-gamma, and gamma-Poisson conjugacy relationships. Efficient closed-form Gibbs updates are detailed in Appendix B for OBSDA inference by adopting novel data augmentation techniques suitable to our model.

4.2.2 SI-OBSDA

We now extend OBSDA to SI-OBSDA, with the goal of incorporating gene-gene network information available *a priori* to have an *informed* inference mechanism. In OBSDA, to be able to derive closed-form updates, we are restricted to certain prior assumptions to take advantage of the appropriate data augmentation and conjugacy relationships. In SI-OBSDA, we want to impose prior constraints stemming from domain knowledge in the inference procedure. Hence, instead of resorting to Gibbs sampling for model inference, in SI-OBSDA we exploit semi-implicit variational inference (SIVI) [111] as the base inference method, which is able to construct flexible variational families to approximate the actual posterior. We first describe the base inference mechanism in SI-OBSDA and then integrate the prior network knowledge.

Denoting the latent variables or parameters of interest as $\boldsymbol{z}$ and the observed data as $\mathbf{x}$ in a general Bayesian model, variational inference maximizes the evidence lower bound (ELBO), defined as

$$
\mathcal{L} = \text{E}_{\boldsymbol{z} \sim q(\boldsymbol{z}|\mathbf{x})} \big[p(\mathbf{x}|\boldsymbol{z}) \big] - \text{KL}\big(q(\boldsymbol{z}|\mathbf{x}) || p(\boldsymbol{z}) \big),
$$

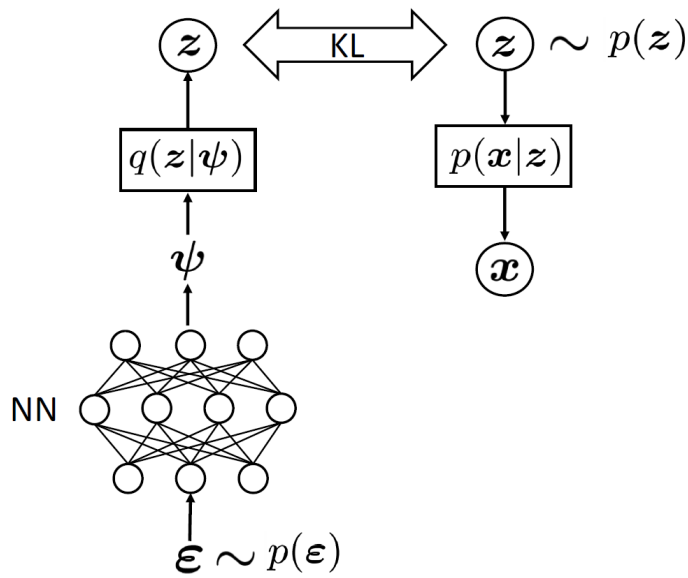


Figure 4.1: Schematic diagram of semi-implicit variational inference in SI-OBSDA

where $q(\boldsymbol{z}|\mathbf{x})$ is the variational posterior selected from a tractable family of distributions and KL denotes the Kullback-Leibler divergence. To simplify the optimization of the ELBO, a commonly adopted choice of variational distributions is the family of factorized distributions. This is referred to as mean-field variational inference (MFVI) [116]. However, MFVI can suffer from various shortcomings, including inability to capture multimodality in the posterior and underestimation of the posterior variance [117].

Here in SI-OBSDA, $\boldsymbol{z}$ denotes the collection of previously described model parameters in OB-SDA, including the association between genes and factors $\{\boldsymbol{\phi}_j\}_{j=1}^{J}$, factors' relevancy to domains and labels $\{\boldsymbol{\theta}_d^l\}_{d=1,l\in L_d}^{D}$, sample variability $\{p_{d,i}^l\}_{d=1,l\in L_d,i=1}^{D,\quad N_d^l}$, label parameters $\{\nu^l\}_{l\in\cup_{d=1}^{D}L_d}$, local factor popularity parameters for each domain $\{u_{d,k}\}_{d=1,k=1}^{D,K}$, global factor $\{b_k\}_{k=1}^{K}$ and domain parameters $\{q_d\}_{d=1}^{D}$, and hyperparameters c_0 and γ_0. We have used L_d, D, and N_d^l to denote the set of labels in domain d, the number of domains, and the number samples in domain d with label l, respectively.

To have more expressive variational families while maintaining computational tractability, in SI-OBSDA we employ SIVI and construct a model with an explicit joint distribution $p(\mathbf{x}, \boldsymbol{z})$ and a semi-implicit approximate posterior $q_{\boldsymbol{\omega}}(\boldsymbol{z})$ (Figure 4.1). In other words, the idea is to define

the variational family in a hierarchical manner as $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{\psi})$, where the conditional variational distribution is explicit but $\boldsymbol{\psi} \sim q_{\boldsymbol{\omega}}(\boldsymbol{\psi})$ is implicit and required to be reparameterizable. More specifically, samples from $q_{\boldsymbol{\omega}}$ can be generated by transforming random noise via a neural network to be more expressive for modeling $\mathbf{x}$. It is clear that the marginal inferred posteriors are not independent as in the standard variational inference, and posterior dependence can be captured.

In SI-OBSDA, we place reparameterizable (location-scale) variational distributions on the parameters. For the parameters in $\mathbb{R}^+$ and $(0, 1)$, we use log-normal (log N) and logistic-normal (logit N) distributions, respectively. For $\{\boldsymbol{\phi}_j\}_{j=1}^J$, in SI-OBSDA we assume logistic-normal prior and variational distributions. This resolves the optimization issue in the simplex while potentially increasing model flexibility. The joint log-likelihood of SI-OBSDA can be found in Appendix B. We place the following reparameterizable variational distributions in our model inference for SI-OBSDA:

$$
\begin{aligned}
q(\boldsymbol{z}|\boldsymbol{\psi}, \boldsymbol{\xi}) = & \prod_{d,l,k} \log \mathrm{N}(\theta_{d,k}^l; \hat{\mu}_{\theta_{d,k}^l}, \hat{\sigma}^2_{\theta_{d,k}^l}) \prod_j \text{logit N}(\boldsymbol{\phi}_j; \hat{\boldsymbol{\mu}}_{\boldsymbol{\phi}_j}, \hat{\Sigma}_{\boldsymbol{\phi}_j}) \\
& \prod_l \log \mathrm{N}(\nu^l; \hat{\mu}_{\nu^l}, \hat{\sigma}^2_{\nu^l}) \prod_{d,k} \log \mathrm{N}(u_{d,k}; \hat{\mu}_{u_{d,k}}, \hat{\sigma}^2_{u_{d,k}}) \\
& \prod_d \log \mathrm{N}(q_d; \hat{\mu}_{q_d}, \hat{\sigma}^2_{q_d}) \prod_k \log \mathrm{N}(b_k; \hat{\mu}_{b_k}, \hat{\sigma}^2_{b_k}) \\
& \prod_{d,l,i} \text{logit N}(p_{d,i}^l; \hat{\mu}_{p_{d,i}^l}, \hat{\sigma}^2_{p_{d,i}^l}) \\
& \log \mathrm{N}(c_0; \hat{\mu}_{c_0}, \hat{\sigma}^2_{c_0}) \log \mathrm{N}(\gamma_0; \hat{\mu}_{\gamma_0}, \hat{\sigma}^2_{\gamma_0}).
\end{aligned} \tag{4.4}
$$

For inference, we optimize an asymptotically exact surrogate evidence lower bound (ELBO) [111]:

$$
\underline{\mathcal{L}}_{\tilde{M}} = \mathrm{E}_{q(\boldsymbol{z}|\boldsymbol{\psi}) q_{\boldsymbol{\omega}}(\boldsymbol{\psi})} \mathrm{E}_{\boldsymbol{\psi}^{(1)}, \cdots, \boldsymbol{\psi}^{(\tilde{M})} \sim q_{\boldsymbol{\omega}}(\boldsymbol{\psi})} \Big[\log \frac{p(\mathbf{x}, \boldsymbol{z}_i)}{\frac{1}{\tilde{M}+1} \big[q(\boldsymbol{z}_i|\boldsymbol{\psi}_i) + \sum_{m=1}^{\tilde{M}} q(\boldsymbol{z}_i|\boldsymbol{\psi}^{(m)}) \big]} \Big], \tag{4.5}
$$

where we have $\lim_{\tilde{M} \to \infty} \underline{\mathcal{L}}_{\tilde{M}} = \text{ELBO}$. In practice, $\boldsymbol{\psi}^{(m)} = T_{\boldsymbol{\omega}}(\boldsymbol{\epsilon}^{(m)})$, where $\boldsymbol{\epsilon}^{(m)} \sim q(\boldsymbol{\epsilon})$, with $q(\boldsymbol{\epsilon})$ being the source of randomness and $T_{\boldsymbol{\omega}}$ a deep neural network (Figure 4.1). The variational

distribution can have additional variational parameters $\boldsymbol{\xi}$, not mixed with another distribution, i.e. we have $q(\boldsymbol{z}|\boldsymbol{\psi}, \boldsymbol{\xi})$. Denoting the reparameterization of $q(\boldsymbol{z}|\boldsymbol{\psi}, \boldsymbol{\xi})$ as $\boldsymbol{z} = f(\boldsymbol{\varepsilon}, \boldsymbol{\xi}, \boldsymbol{\psi})$, $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$, where $p(\boldsymbol{\varepsilon})$ is the source of randomness, $\boldsymbol{z}$ can be sampled by $\boldsymbol{z}_i = f(\boldsymbol{\varepsilon}_i, \boldsymbol{\xi}, \boldsymbol{\psi}_i)$, $\boldsymbol{\varepsilon}_i \sim p(\boldsymbol{\varepsilon})$. The parameters of the mixing distribution and the variational parameters can be optimized by gradient ascent:

$$\begin{aligned} \boldsymbol{\xi} &= \boldsymbol{\xi} + \rho_t \nabla_{\boldsymbol{\xi}} \underline{\mathcal{L}}_{\tilde{M}}\Big(\{\boldsymbol{\psi}^{(m)}\}, \{\boldsymbol{\psi}_i\}, \{\boldsymbol{z}_i\}\Big), \\ \boldsymbol{\omega} &= \boldsymbol{\omega} + \upsilon_t \nabla_{\boldsymbol{\omega}} \underline{\mathcal{L}}_{\tilde{M}}\Big(\{\boldsymbol{\psi}^{(m)}\}, \{\boldsymbol{\psi}_i\}, \{\boldsymbol{z}_i\}\Big). \end{aligned} \tag{4.6}$$

In SI-OBSDA, we consider the collection of $\{\hat{\mu}_{\theta^l_{d,k}}\}_{d=1,l\in L_d,k=1}^{D,\quad K}$, $\{\hat{\boldsymbol{\mu}}_{\boldsymbol{\phi}_j}\}_{j=1}^{J}$, $\{\hat{\mu}_{\nu^l}\}_{l\in\cup_{d=1}^{D} L_d}$, $\{\hat{\mu}_{b_k}\}_{k=1}^{K}$, $\{\hat{\mu}_{u_{d,k}}\}_{d=1,k=1}^{D,K}$, $\{\hat{\mu}_{q_d}\}_{d=1}^{D}$, $\hat{\mu}_{c_0}$, and $\hat{\mu}_{\gamma_0}$ to be the parameters governed by the mixing distribution of $\boldsymbol{\psi}$, and $\{\hat{\mu}_{p^l_{d,i}}\}_{d=1,l\in L_d,i=1}^{D,\quad N^l_d}$, $\{\hat{\sigma}_{p^l_{d,i}}\}_{d=1,l\in L_d,i=1}^{D,\quad N^l_d}$, $\{\hat{\sigma}_{\theta^l_{d,k}}\}_{d=1,l\in L_d,k=1}^{D,\quad K}$, $\{\hat{\Sigma}_{\boldsymbol{\phi}_j}\}_{j=1}^{J}$, $\{\hat{\sigma}_{\nu^l}\}_{l\in\cup_{d=1}^{D} L_d}$, $\{\hat{\sigma}_{b_k}\}_{k=1}^{K}$, $\{\hat{\sigma}_{u_{d,k}}\}_{d=1,k=1}^{D,K}$, $\{\hat{\sigma}_{q_d}\}_{d=1}^{D}$, $\hat{\sigma}_{c_0}$, and $\hat{\sigma}_{\gamma_0}$ as the variational parameters ($\boldsymbol{\xi}$). For numerical stability we further reparameterize the variational parameters by log-transform and Cholesky factorization. Implementation details of SI-OBSDA is included in Appendix B.

In SI-OBSDA, similar to the SIVI inference mechanism in [111], we employ a neural network as $T_{\boldsymbol{\omega}}$ for the mixing distribution. Since neural networks have high modeling capacity, $q_{\boldsymbol{\omega}}(\boldsymbol{\psi})$ can be highly flexible, and the dependencies between the elements of $\boldsymbol{\psi}$ can be well captured. Moreover, from the implementation perspective, neural networks can easily leverage automatic differentiation to optimize the surrogate ELBO in (4.5), which is computationally desirable.

4.2.3 Incorporating Prior Network Knowledge in SI-OBSDA

In addition to the expression data, there exists *a priori* interactome knowledge such as gene-gene interaction network that contains genome-scale connectivity information [85]. These can be derived based on either regulatory, metabolic, signaling interactions, or protein binding.

In SI-OBSDA we impose constraints stemming from the prior knowledge in the gene-factor associations to construct informed latent representations and inference. More specifically, since

the factors can be interpreted as functional modules or underlying biological processes, intuitively, the genes that are connected in the prior knowledge network should have closer associations to the underlying factors. Hence, we impose proximity constraints on the variables quantifying the association between genes and factors for genes that are connected in the prior knowledge network. Specifically, we add a regularization term coming from prior belief to the objective of the SI-OBSDA:

$$\begin{aligned} &\mathcal{L}_{\text{SI-OBSDA}} = \underline{\mathcal{L}}_{\tilde{M}} + \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\psi},\boldsymbol{\xi})}\mathcal{L}_{\text{pr}}, \\ &\text{where} \quad \mathcal{L}_{\text{pr}} = \sum_{j=1}^{J} \sum_{\tilde{j}\in\mathcal{C}_j, \tilde{j}<j} \lambda_{j,\tilde{j}} ||\boldsymbol{\phi}_j - \boldsymbol{\phi}_{\tilde{j}}||. \end{aligned} \tag{4.7}$$

In the equation above, $\mathcal{C}_j$ denotes the set of genes that are connected to gene j in the prior network knowledge.

The proposed additive constraints when optimizing for inference fit in the MKDIP prior-construction framework of [88], with the expectation taken over the variational distribution. More specifically, we can consider slackness for the prior constraints which are linearly added to the objective, i.e. the regularization term acts as a relaxation of the constraints coming from prior knowledge with $\lambda_{j,\tilde{j}}$ encoding the degree of belief in the specific prior interaction edge. In other words, the higher the confidence in an edge is in prior knowledge, the larger $\lambda_{j,\tilde{j}}$ will be set.

Another way to interpret the regularization term is through assuming (conditional) prior distributions that impose these constraints in effect. Moreover, although different in nature, it is worth noting that our work has connections with recent works including [118], where additional label information is imposed through proximity constraints in the latent space and has been shown to be beneficial even on large data.

4.2.4 Classification with OBSDA and SI-OBSDA

In the previous sections, we have introduced the models and inference procedures for OBSDA and SI-OBSDA. Here, we describe how classification for subtyping is done based on the inferred Bayesian models. The classification operator in OBSDA and SI-OBSDA is based on the optimal

Bayesian classification (OBC) framework [109, 43, 119]. In OBC, the design of the classifier is based on the posterior marginalization of the class-conditional feature distributions, called effective class-conditional distributions. This is in contrast to *plug-in* classifier design where the estimates of the parameters are used to calculate the class-conditional distributions to form the classifier, which may not result in the optimal expected error relative to the posterior distributions, especially when the posteriors are multi-modal. More specifically, denoting the collection of all model parameters and the posteriors after observing data as Θ and π^*, respectively, OBC classifier (f_{obc}) satisfies

$$\mathbb{E}_{\pi^*}[\delta(f_{\text{obc}}, \Theta)] \leq \mathbb{E}_{\pi^*}[\delta(f, \Theta)], \qquad \forall f \in F, \tag{4.8}$$

where f and F denote a classifier and all classifiers possessing measurable decision regions, respectively; and $\delta(\cdot, \cdot)$ is the error for fixed parameter values and a classification rule.

In OBSDA and SI-OBSDA, we can derive the optimal Bayesian classifier in the target domain (OBTD) based on the samples of the parameters of the target domain generated in the inference chain of OBSDA or from the variational posteriors in SI-OBSDA. Note that this is equivalent to marginalizing the joint posterior over the source domain(s) as in [119].

Denoting the class prior probabilities in the target domain ($d = t$, and without loss of generality assuming the labels are from 1 to L_t) as $\boldsymbol{c}_t = (c_t^1, \cdots, c_t^{L_t})$, and given the parameters of the model, the probability of sample $\mathbf{x}_{t,i}$ belonging to class l is equal to

$$p(l|\mathbf{x}_{t,i}) = \frac{c_t^l p(\mathbf{x}_{t,i}|\boldsymbol{\Phi}, \boldsymbol{\theta}_t^l, p_{t,i}^l)}{\sum_{\tilde{l}=1}^{L_t} c_t^{\tilde{l}} p(\mathbf{x}_{t,i}|\boldsymbol{\Phi}, \boldsymbol{\theta}_t^{\tilde{l}}, p_{t,i}^{\tilde{l}})}, \tag{4.9}$$

where $p(\mathbf{x}_{t,i}|\boldsymbol{\Phi}, \boldsymbol{\theta}_t^l, p_{t,i}^l) = \prod_{j=1}^{J} \text{NB}(\mathbf{x}_{t,j,i}|\boldsymbol{\phi}_j^T \boldsymbol{\theta}_t^l, p_{t,i}^l)$. Hence, the optimal Bayesian classifier in the target domain (OBTD) is:

$$f_{\text{OBTD}}(\mathbf{x}_{t,i}) = \arg\max_{l \in \{1, \cdots, L_t\}} \mathrm{E}_{\pi^*}\left[c_t^l p(\mathbf{x}_{t,i}|\boldsymbol{\Phi}, \boldsymbol{\theta}_t^l, p_{t,i}^l)\right]. \tag{4.10}$$

Assuming that the class prior probabilities in the target domain are independent of the other model

parameters *a priori* and have a Dirichlet prior $(c_t^1, \cdots, c_t^{L_t}) \sim \text{Dir}(\eta_t^1, \cdots, \eta_t^{L_t})$, we have

$$f_{\text{OBTD}}(\mathbf{x}_{t,i}) = \arg\max_{l \in \{1,\cdots,L_t\}} \mathbb{E}_{\pi^*}\left[c_t^l\right] \mathbb{E}_{\pi^*}\left[p(\mathbf{x}_{t,i}|\mathbf{\Phi}, \boldsymbol{\theta}_t^l, p_{t,i}^l)\right], \tag{4.11}$$

where

$$\mathbb{E}_{\pi^*}\left[c_t^l\right] = \frac{|\mathbf{x}_t^l| + \eta_t^l}{\sum_{\tilde{l}=1}^{L_t} |\mathbf{x}_t^{\tilde{l}}| + \eta_t^{\tilde{l}}}. \tag{4.12}$$

$|\mathbf{x}_t^l|$ denotes the number of training samples in the target domain t with label l.

Given the training data, OBSDA generates samples from the posteriors of the parameters via the Gibbs chain. Similarly, in SI-OBSDA when the optimization of the training loss is stopped, samples from the posterior can be generated by pushing random noise samples through the trained neural network and in turn using the outputs as parameters for sampling from the variational posteriors. We collect these samples (or save the neural network in SI-OBSDA) in the training procedure and use them at test time. When a new unlabeled test data i comes in, we only need to generate posterior samples for $p_{t,i}^l$ corresponding to the collected posterior samples for $\boldsymbol{\theta}_t^l$ by (B.12) in Appendix B to predict the label for the data point by (4.10).

4.3 Results and Discussion

4.3.1 Data

We evaluate the performance of our OBSDA and SI-OBSDA for subtyping lung cancer using several RNA-Seq datasets from The Cancer Genome Atlas (TCGA) [120]. In our experiments, we consider RNA-Seq data from two subtypes of non-small cell lung cancer (NSCLC), lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC) as the target domain. According to the American Cancer Society statistics, lung cancer is the second most commonly diagnosed cancer and the leading cause of cancer death in both men and women in the United States. About 84% of lung cancers are NSCLC and LUAD and LUSC combined account for about 70% of lung cancers.

We examine the target lung cancer subtyping accuracy by ours and other competing methods,

focusing on evaluating their performances when using additional RNA-Seq data from three different source domains that either share the same class labels with or have different ones from the target domain. Specifically, we take RNA-SeqV2 dataset, which is from the second analysis pipeline, for LUAD and LUSC as the first source domain, RNA-Seq data from Head and Neck Squamous Cell Carcinoma (HNSC) as the second source domain, and data from the two most common types of kidney cancers, kidney renal clear cell carcinoma (KIRC) and kidney renal papillary cell carcinoma (KIRP) as the third source domain. Clearly, the degree to which the source domain may help lung cancer subtyping vary for these three different source domains. One is from the data with the same subtypes but different NGS pipelines, while the other two are from studies concerning different cancer types with one and two classes in each domain.

For SI-OBSDA we use the gene-gene network containing only physical interactions (the human interactome) archived in [85] as the network prior knowledge. The network, which features 13460 proteins interconnected by 141296 interactions, does not include interactions extracted from gene expression data, and has been compiled by combining experimental support from several databases including protein-protein and regulatory interactions, signaling interactions, metabolic pathway interactions, and kinase-substrate interactions. In the experiments, we consider equal weights for the edges in SI-OBSDA, and set them to either 1 or 0.25 based on the accuracy of the inferred model on the training data. For SI-OBSDA, in all the experiments we take $\boldsymbol{\epsilon}$ to have the same cardinality of $\boldsymbol{\psi}$, and $T_{\boldsymbol{\omega}}(\boldsymbol{\epsilon})$ as a neural network with three hidden layers (more implementation details available in Appendix B).

In the following experiments, we first pick the common genes within the target and source datasets and the prior network knowledge, resulting in 11839 genes. We then remove the genes that have total read counts of less than 40 across the LUAD and LUSC samples in the target domain. Finally, we perform differential expression analysis with DESeq2 [121] and select 500 out of the top 2500 genes with the highest log-fold change (with gaps of 5) in each experimental run for all the methods for fair comparison.

4.3.2 Baselines

As the baselines for comparing lung cancer subtyping accuracy, we apply SVM (with both Gaussian and linear kernels), regularized linear SVM, and regularized logistic regression on the data from the target domain. We also use a neural network (NN) classifier as an additional baseline. The architecture of the network is kept the same as the neural network utilized in the inference mechanism of SI-OBSDA (explained in detail in Appendix B) to have a fair comparison for evaluating the proposed models. The only architectural difference is that the NN classifier takes the expression data as input and outputs the logit (log-odds). In the first setup with the source domain having the same labels as the target domain, we train these baselines once only using the training data in the target domain, and once using the collection of source and target training data. We tune the hyperparameters of each baseline classifier in each run given the training data with Bayesian optimization [122, 123] and the cross-validation loss as the objective function.

To compare the performance of our method in terms of domain adaptation and learning useful information from source domains for designing a predictor in the target domain, there are two other methods that can provide good comparisons that can be applied for domain adaptation and transfer learning on NGS count data for comparisons. Optimal Bayesian transfer learning (OBTL) [119, 108] is a supervised transfer learning method that models the relationship between the same classes across domains by assuming joint priors and marginalizing the joint posterior over the source domain parameters. Unfortunately, this method is not scalable to more than 10 to 20 genes, so we could not perform comparisons with it. BMDL [107] is a multi-domain learning method that projects the data from different domains to a lower dimensional common embedding space, and applies a classifier on the projected space. It has been shown that BMDL outperforms other similar Bayesian latent models on the NGS classification problem. Thus, we choose BMDL as the state-of-the-art baseline for our experiments on domain adaption for RNA-Seq data.

4.3.3 Results

4.3.3.1 LUAD and LUSC data in source and target domains

In this setup, we compare the performance of different methods when the source and target domains have data from the same cancer subtypes. The target domain contains 162 and 240 samples from LUAD and LUSC, respectively. In each run, we randomly pick 20 samples in total from the target domain for training by stratified sampling, and use the rest of the samples in the target domain for testing. The source domain contains 414 and 312 samples from LUAD and LUSC, respectively, where we perform stratified sampling (considering the source proportions) for different number of training samples from the source domain. We investigate the performance of OBSDA, BMDL, regularized logistic regression (Reg Log), regularized linear SVM (Reg SVM), kernel SVM (SVM), and neural network classifier (NN) using three different numbers of source samples, 564, 112, and 11. This setup covers a wide range of source samples, from a few training samples from source (nearly half of target training samples) to around 5.5$\times$ and 28$\times$ the number of target samples in the training data. Note that in this experiment, since the labels are the same across domains, we train the single-domain baseline methods once utilizing the collection of all the training data from both domains and once only the target domain's training data.

The results in Figure 4.2 show that OBSDA achieves the best performance compared with the baselines by effectively borrowing information from the source data. We can see that OBSDA's error in classifying subtypes in the target domain consistently decreases as the number of source samples increases. On the contrary, BMDL seems to suffer when the source samples drastically dominate the target samples in the training data, which is undesirable for domain adaptation. We can also observe this adverse effect of having a lot more source samples than target samples in the training data on the NN classifier, where the results show that the proposed methods outperform the NN classifier for all the numbers of source samples. This confirms that neural networks are not specifically fit to use on smaller datasets and indicates that explicitly modeling for learning useful information from other domains for the target domain is required when facing smaller (target)

sample sizes.

Next, we test the performance of SI-OBSDA that incorporates constraints on the latent space stemming from the prior knowledge within a flexible variational inference in this experiment setup. As seen in Figure 4.2, similar to OBSDA, SI-OBSDA's error also consistently decreases as the number of source samples increases. The results in Table 4.1 show around 1% to 3% improvement compared with OBSDA and 4% to 5% difference from BMDL, demonstrating that SI-OBSDA can achieve the best performance by incorporating prior knowledge as well as learning useful information across domains.

It is worth noting that SI-OBSDA and OBSDA also show relatively lower variance across the experimental runs, i.e. a more robust performance, compared with the other methods.

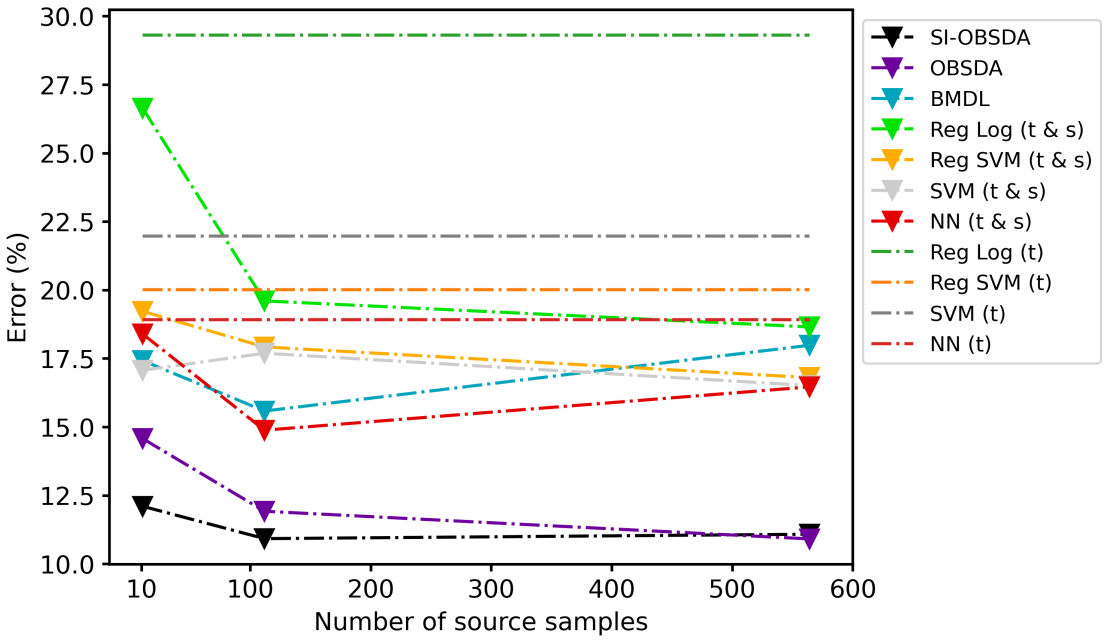


Figure 4.2: Average performance of different methods in identifying cancer subtypes of LUAD vs LUSC using different number of source samples. (t) and (t & s) correspond to using only target samples, and source and target samples in training, respectively.

Table 4.1: Average errors (in % $\pm$ standard deviations) in identifying subtypes of LUAD vs LUSC with the source domain containing samples from the same subtypes.

Method	$N_s = 11$	$N_s = 112$
SI-OBSDA	12.10 ± 0.81	10.92 ± 0.47
OBSDA	14.57 ± 0.64	11.91 ± 1.09
BMDL	17.42 ± 1.66	15.58 ± 1.19
Reg Log (t & s)	26.63 ± 2.92	19.60 ± 3.18
Reg SVM (t & s)	19.22 ± 5.64	17.92 ± 1.56
SVM (t & s)	17.07 ± 4.53	17.69 ± 1.23
NN (t & s)	18.39 ± 3.63	14.89 ± 1.33
Reg Log (t)	29.31 ± 4.41	29.31 ± 4.41
Reg SVM (t)	20.01 ± 2.57	20.01 ± 2.57
SVM (t)	21.97 ± 2.67	21.97 ± 2.67
NN (t)	18.91 ± 3.26	18.91 ± 3.26

4.3.3.2 LUAD and LUSC data only in the target domain

In this section, we examine the performance of different methods using data from source domains that do not have labels in common with the data from the target domain. We consider HNSC data as one source domain and kidney cancer data (KIRC and KIRP) as another source domain. The HNSC dataset contains 294 samples, and the kidney cancer dataset consists of 537 KIRC and 14 KIRP samples. We have selected these datasets from different cancer types as the source domain since the degree to which they may help detecting the lung cancer subtypes may be different due to the different disease mechanisms. Moreover, another difference is the number of labels in each source domain with one domain only containing data with one label (HNSC), and the other containing data with two labels (KIRC and KIRP). Similar to the previous section, in each Monte Carlo run we do stratified sampling for training data from the target domain, randomly picking 20 training samples from the target domain. For the lower and higher number of source samples ($N_S = 11$ and $N_s = 112$), two random or all the 14 KIRP samples are selected for training, respectively, with the rest of the source training samples coming from KIRC.

The results in Table 4.2 demonstrate that both SI-OBSDA and OBSDA outperform BMDL when the source domain contains data of different cancers from the target domain by close to 5%

to 7% under different settings. We can attribute this to BMDL not leveraging label information in the latent representation learning stage. Comparing the numbers in Tables 4.2 and 4.1, we see that all the methods that use data from both source and target domains still perform better than the other baselines using only the target domain data in training. Similar to the previous experiment, SI-OBSDA, which leverages the prior network knowledge in addition to the expression data within its flexible variational inference, achieves the best accuracy in classifying subtypes in the target domain. It is interesting to note that OBSDA and SI-OBSDA both benefit from more samples from the source domain in training, even though they are from different cancer types. This verifies the benefit of our proposed approach in modeling that can borrow useful information from other domains and labels for the prediction task in the target domain. Also, the results in Tables 4.2 and 4.1 show that, as expected, when the source contains data from the same labels as the target domain, SI-OBSDA and OBSDA generally achieve better accuracy for the same number of source samples used in training. Additionally, when the data from the source are for different cancers from the target domain, the decrease in prediction error in the target domain is slower when increasing the number of source samples, compared with the case of source domain containing data from the same disease.

Table 4.2: Average errors (in % $\pm$ standard deviations) in identifying subtypes of LUAD vs LUSC with the source domain containing samples from different labels.

Source sample size	$N_s = 11$	$N_s = 112$
Source domain	HNSC	
SI-OBSDA	12.56 ± 0.87	11.85 ± 0.77
OBSDA	13.48 ± 0.95	13.02 ± 0.47
BMDL	17.32 ± 3.38	17.75 ± 3.13
Source domain	KIRC,KIRP	
SI-OBSDA	12.17 ± 0.88	12.23 ± 0.65
OBSDA	14.59 ± 1.70	14.20 ± 0.67
BMDL	19.81 ± 1.76	17.82 ± 2.33

4.3.3.3 *Effect of incorporating prior knowledge*

The results in the previous experiments showed that SI-OBSDA, which takes advantage of flexible variational posteriors and the gene-gene network prior knowledge, outperforms OBSDA and the baselines. Here, we examine the effect of the incorporation of the constraints coming from prior knowledge within the inference optimization on the performance of SI-OBSDA. Table 4.3 shows the results of SI-OBSDA with and without using prior knowledge for the different settings of source domain and number of source samples. The results suggest that SI-OBSDA generally benefits from the prior network knowledge by varying degrees for different setups. Note that by comparing the numbers in Table 4.3 with the numbers in Tables 4.1 and 4.2, we see that without incorporating the prior constraints on the latent space, SI-OBSDA attains errors that are still comparable or slightly lower than OBSDA in most cases while being better than BMDL by 4% to 7%.

Table 4.3: Comparison of SI-OBSDA and SI-OBSDA without prior knowledge (SI-OBSDA w/o Prior) in terms of average errors (in %) in identifying subtypes of LUAD vs LUSC with different source domain settings.

Method		**SI-OBSDA**	**SI-OBSDA w/o Prior**
Lung source data	$N_s = 11$	12.10	13.09
	$N_s = 112$	10.92	12.04
HNSC source data	$N_s = 11$	12.56	13.28
	$N_s = 112$	11.85	12.83
Kidney source data	$N_s = 11$	12.17	12.90
	$N_s = 112$	12.23	13.02

5. OPTIMAL CLUSTERING WITH MISSING VALUES *

5.1 Introduction

Missing values frequently arise in modern biomedical studies due to various reasons, including missing tests or complex profiling technologies for different omics measurements. Missing values can complicate the application of clustering algorithms, whose goals are to group points based on some similarity criterion. A common practice for dealing with missing values in the context of clustering is to first impute the missing values, and then apply the clustering algorithm on the completed data. In this Chapter, we consider missing values in the context of optimal clustering, which finds an optimal clustering operator with reference to an underlying random labeled point process (RLPP). We show how the missing-value problem fits neatly into the overall framework of optimal clustering by incorporating the missing value mechanism into the random labeled point process and then marginalizing out the missing-value process.

Clustering has been a mainstay of genomics since the early days of gene-expression microarrays [124]. For instance, expression profiles can be taken over various tissue samples and then clustered according to the expression levels for each sample, the aim being to discriminate pathologies based on their differential patterns of gene expression [125]. In particular, model-based clustering, which assumes that the data are generated by a finite mixture of underlying probability distributions, has gained popularity over heuristic clustering algorithms, for which there is no concrete way of determining the number of clusters or the best clustering method [126]. Model-based clustering methods [127] provide more robust criteria for selecting the appropriate number of clusters. For example, in a Bayesian framework, utilizing Bayes Factor can incorporate both *a priori* knowledge of different models, and goodness of fit of the parametric model to the observed data. Moreover, nonparametric models such as Dirichlet-process mixture models [128] provide a more flexible approach for clustering, by automatically learning the number of components. In small-sample

*Reprinted with permission from S. Boluki, S. Z. Dadaneh, X. Qian, and E. R. Dougherty, “Optimal clustering with missing values," BMC Bioinformatics, vol. 20, no. 12, pp. 1–10, 2018. Copyright 2018 Authors.

settings, model-based approaches that incorporate model uncertainty have proved successful in designing robust operators [43], as also seen in the previous Chapters, and in objective-based experiment design to expedite the discovery of such operators [129, 130].

Whereas classification theory is grounded in feature-label distributions with the error being the probability that the classifier mislabels a point [43]; clustering algorithms operate on random labeled point processes (RLPPs) with error being the probability that a point will be placed into the wrong cluster (partition) [131]. An optimal (Bayes) clusterer minimizes the clustering error and can be found with respect to an appropriate representation of the cluster error [132].

A common problem in clustering is the existence of missing values. These are ubiquitous with high-throughput sequencing technologies, such as microarrays [133] and RNA sequencing (RNA-seq) [134]. For instance, with microarrays, missing data can occur due to poor resolution, image corruption, or dust or scratches on the slide [135], while for RNA-seq, the sequencing machine may fail to detect genes with low expression levels owing to the random sampling nature of sequencing technologies. As a result of these missing data mechanisms, gene expression data from microarray or RNA-seq experiments are usually in the form of large matrices, with rows and columns corresponding to genes and experimental conditions or different subjects, respectively, with some values missing. Imputation methods, such as *MICE* [136], *Amelia II* [137] and *missForest* [138], are usually employed to complete the data matrix before clustering analysis; however, in small-sample settings, which are common in genomic applications, these methods face difficulties, including co-linearity due to potential high correlation between genes in samples, which precludes the successful imputation of missing values.

In this Chapter we follow a different direction by incorporating the generation of missing values with the original generating random labeled point process, thereby producing a new RLPP that generates the actual observed points with missing values. The optimal clusterer in the context of missing values is obtained by marginalizing out the missing features in the new RLPP. One potential challenge arising here is that in the case of missing values with general patterns, conducting the marginalization can be computationally intractable, and hence resorting to approximation methods

such as Monte Carlo integration is necessary.

Although the proposed framework for optimal clustering can incorporate the probabilistic modeling of arbitrary types of missing data mechanisms, to facilitate analysis, throughout this work we assume data are missing completely at random (MCAR) [139]. In this scenario, the parameters of the missingness mechanism are independent of other model parameters and therefore vanish after the expectation operation in the calculation of the posterior of label functions for clustering assignment.

We derive the optimal clusterer for different scenarios in which features are distributed according to multivariate Gaussian distributions. The performance of this clusterer is compared to various methods, including k-POD [140] and fuzzy c-means with optimal completion strategy [141], which are methods for directly clustering data with missing values, and also k-means [142], fuzzy c-means [143] and hierarchical clustering [144] with the missing values imputed. Comprehensive simulations based on synthetic data show the superior performance of the proposed framework for clustering with missing values over a range of simulation setups. Moreover, evaluations based on RNA-seq data further verify the superior performance of the proposed method in a real-world application with missing data.

5.2 Methods

5.2.1 Optimal Clustering

Given a point set $S \subset \mathbb{R}^d$, where d is the dimension of the space, denote the number of points in S by $\eta(S)$. A *random labeled point process* (RLPP) is a pair (Ξ, Λ), where Ξ is a point process generating S and Λ generates random labels on point set S. Ξ maps from a probability space to $[N; \mathcal{N}]$, where N is the family of finite sequences in $\mathbb{R}^d$ and $\mathcal{N}$ is the smallest σ-algebra on N such that for any Borel set B in $\mathbb{R}^d$, the mapping $S \to \eta(S \cap B)$ is measurable. A random labeling is a family, $\Lambda = \{\Phi_S : S \in N\}$, where Φ_S is a random label function on the point set S in N. Denoting the set of labels by $L = \{1, 2, ..., l\}$, Φ_S has a probability mass function on L^S defined by $P_S(\phi_S) = P(\Phi_S = \phi_S | \Xi = S)$, where $\phi_S : S \to L$ is a deterministic function assigning a label

to each point in S.

A label operator λ maps point sets to label functions, $\lambda(S) = \phi_{S,\lambda} \in L^S$. For any set S, label function ϕ_S and label operator λ, the *label mismatch error* is defined as

$$\epsilon_\lambda(S, \phi_S) = \frac{1}{\eta(S)} \sum_{x \in S} I_{\phi_S(x) \neq \phi_{S,\lambda}(x)}, \tag{5.1}$$

where I_A is an indicator function equal to 1 if A is true and 0 otherwise. The *error of label function* $\lambda(S)$ is computed as $\epsilon_\lambda(S) = \mathbb{E}_{\Phi_S}[\epsilon_\lambda(S, \phi_S)|S]$, and the *error of label operator* λ for the corresponding RLPP is then defined by $\epsilon[\lambda] = \mathbb{E}_\Xi \mathbb{E}_{\Phi_\Xi}[\epsilon_\lambda(\Xi, \phi_\Xi)]$.

Clustering involves identifying partitions of a point set rather than the actual labeling, where a partition of S into l clusters has the form $\mathcal{P}_S = \{S_1, S_2, ..., S_l\}$ such that S_i's are disjoint and $S = \bigcup_{i=1}^{l} S_i$. A cluster operator ζ maps point sets to partitions, $\zeta(S) = \mathcal{P}_{S,\zeta}$. Considering the label switching property of clustering operators, let us define F_ζ as the family of label operators that all induce the same partitions as the clustering operator ζ. More precisely, a label function ϕ_S induces partition $\mathcal{P}_S = \{S_1, S_2, ..., S_l\}$, if $S_i = \{x \in S : \phi_S(x) = l_i\}$ for distinct $l_i \in L$. Thereby, $\lambda \in F_\zeta$ if and only if $\phi_{S,\lambda}$ induces the same partition as $\zeta(S)$ for all $S \in N$. For any set S, label function ϕ_S and cluster operator ζ, define the *cluster mismatch error* by

$$\epsilon_\zeta(S, \phi_S) = \min_{\lambda \in F_\zeta} \epsilon_\lambda(S, \phi_S), \tag{5.2}$$

the *error of partition* $\zeta(S)$ by $\epsilon_\zeta(S) = \mathbb{E}_{\Phi_S}[\epsilon_\zeta(S, \phi_S)|S]$ and the *error of cluster operator* ζ for the RLPP by $\epsilon[\zeta] = \mathbb{E}_\Xi \mathbb{E}_{\Phi_\Xi}[\epsilon_\zeta(\Xi, \phi_\Xi)]$.

As shown in [132], error definitions for partitions can be represented in terms of risk with intuitive cost functions. Specifically, define $G_{\mathcal{P}_S}$ such that $\phi_S \in G_{\mathcal{P}_S}$ if and only if ϕ_S induces $\mathcal{P}_S$. The error of partition can be expressed as

$$\epsilon_\zeta(S) = \sum_{\mathcal{P}_S \in \mathcal{K}_S} c_S(\zeta(S), \mathcal{P}_S) P_S(\mathcal{P}_S), \tag{5.3}$$

where $\mathcal{K}_S$ is the set of all possible partitions of S, $P_S(\mathcal{P}_S) = \sum_{\phi_S \in G_{\mathcal{P}_S}} P_S(\phi_S)$ is the probability

mass function on partitions $\mathcal{P}_S$ of S, and the *partition cost function* between partitions $\mathcal{P}_S$ and $\mathcal{Q}_S$ of S is defined as

$$c_S(\mathcal{Q}_S, \mathcal{P}_S) = \frac{1}{\eta(S)} \min_{\phi_{S,\mathcal{Q}_S} \in G_{\mathcal{Q}_S}} \sum_{x \in S} I_{\phi_{S,\mathcal{P}_S} \neq \phi_{S,\mathcal{Q}_S}}, \tag{5.4}$$

with $\phi_{S,\mathcal{P}_S}$ being any member of $G_{\mathcal{P}_S}$. A Bayes cluster operator ζ^* is a clusterer with the minimal error $\epsilon[\zeta^*]$, called the *Bayes error*, obtained by a Bayes partition, $\zeta^*(S)$ for each set $S \in N$:

$$\begin{aligned} \zeta^*(S) &= \arg \min_{\zeta(S) \in \mathcal{K}_S} \epsilon_\zeta(S) \\ &= \arg \min_{\zeta(S) \in \mathcal{K}_S} \sum_{\mathcal{P}_S \in \mathcal{K}_S} c_S(\zeta(S), \mathcal{P}_S) P_S(\mathcal{P}_S). \end{aligned} \tag{5.5}$$

The Bayes clusterer can be solved for each fixed S individually. More specifically, the search space in the minimization and the set of partitions with known probabilities in the summation can be constrained to subsets of $\mathcal{K}_S$, denoted by $\mathcal{C}_S$ and $\mathcal{R}_S$, respectively. We refer to $\mathcal{C}_S$ and $\mathcal{R}_S$ as the set of candidate partitions and the set of reference partitions, respectively. We can search for the optimal clusterer based on both optimal and suboptimal procedures with derived bounds that can be used to optimally reduce the size of $\mathcal{C}_S$ and $\mathcal{R}_S$.

5.2.2 Gaussian Model with Missing Values

We consider an RLPP model that generates the points in the set S according to a Gaussian model, where features of $x \in S$ can be missing completely at random due to a missing data mechanism independent of the RLPP. More precisely, the points $x \in S$ with label $\phi_S(x) = i$ are drawn independently from a Gaussian distribution with parameters $\rho_i = \{\mu_i, \Sigma_i\}$. Assuming n_i sample points with label i, we divide the observations into $G_i \leq n_i$ groups, where all n_{ig} points in group g have the same set, J_{ig}, of observed features with cardinality $|J_{ig}| = d_{ig}$. Denoting by S_{ig} the set of sample points in group g of label i, we represent the pattern of missing data in this

group using a $d_{ig} \times d$ matrix M_{ig}, where each row is a d-dimensional vector with a single non-zero element with value 1 corresponding to the observed feature's index. Thus, the non-missing portion of sample point $x \in S_{ig}$, i.e. $M_{ig}x$, has Gaussian distribution $\mathrm{N}(M_{ig}\mu_i, M_{ig}\Sigma_i M_{ig}^T)$.

Given $\rho = \{\rho_1, \rho_2, ..., \rho_l\}$ of independent parameters, to evaluate the posterior probability of random labeling function $\phi_S \in L^S$, we have

$$
\begin{aligned}
&P_S(\phi_S) \propto P(\phi_S) f(S|\phi_S) = P(\phi_S) \int f(S|\phi_S, \rho) f(\rho) d\rho = \\
&P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \int \Big(\prod_{x \in S_i} f_i(x|\rho_i) \Big) f(\rho_i) d\rho_i = \\
&P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \int \Big(\prod_{g=1}^{G_i} \prod_{x \in S_{ig}} \mathrm{N}\big(M_{ig}x; M_{ig}\mu_i, M_{ig}\Sigma_i M_{ig}^T\big) \Big) f(\mu_i, \Sigma_i) d\mu_i d\Sigma_i,
\end{aligned}
\tag{5.6}
$$

where $P(\phi_S)$ is the prior probability on label functions, which we assume does not depend on the specific points in S.

5.2.2.1 Gaussian means and known covariances

Under this model, data points are generated according to Gaussians whose mean parameters are random and their covariance matrices are fixed. Specifically, for label i we have $\mu_i \sim \mathrm{N}(m_i, \frac{1}{\nu_i}\Sigma_i)$, where $\nu_i > 0$ and m_i is a length d real vector. Thus the posterior of label function given the point set S can be derived as

$$
\begin{aligned}
&P_S(\phi_S) \propto P(\phi_S) \prod_{\substack{i=1 \\ n_i \geq 1}}^{l} \Bigg[\prod_{g=1}^{G_i} \Big[|2\pi\Sigma_{ig}|^{-n_{ig}/2} \times \exp\{-\frac{1}{2}\mathrm{tr}\big(\Psi_{ig}(\Sigma_{ig})^{-1}\big)\} \Big] \times (\nu_i)^{d/2} |2\pi\Sigma_i|^{-1/2} \\
&\int \exp\Big\{ -\frac{1}{2} \sum_{g=1}^{G_i} n_{ig}(m_{ig} - M_{ig}\mu_i)^T (\Sigma_{ig})^{-1} (m_{ig} - M_{ig}\mu_i) - \frac{\nu_i}{2}(\mu_i - m_i)^T \Sigma_i^{-1} (\mu_i - m_i) \Big\} d\mu_i \Bigg].
\end{aligned}
\tag{5.7}
$$

By completing the square and using the normalization constant of multivariate Gaussian distri-

bution, the integral in this equation can be expressed as

$$\int \exp\Big\{ -\frac{1}{2}\big[(\mu_i - A_i^{-1}b_i)^T A_i(\mu_i - A_i^{-1}b_i) + \sum_{g=1}^{G_i} n_{ig} m_{ig}^T \Sigma_{ig}^{-1} m_{ig} + \nu_i m_i^T \Sigma_i^{-1} m_i - b_i^T A_i^{-1} b_i\big]\Big\}$$
$$= |A_i/(2\pi)|^{-1/2} \exp\Big\{ -\frac{1}{2}\big[\sum_{g=1}^{G_i} n_{ig} m_{ig}^T \Sigma_{ig}^{-1} m_{ig} + \nu_i m_i^T \Sigma_i^{-1} m_i - b_i^T A_i^{-1} b_i\big]\Big\},$$

where

$$A_i = \sum_{g=1}^{G_i} n_{ig} M_{ig}^T \Sigma_{ig}^{-1} M_{ig} + \nu_i \Sigma_i^{-1}, \tag{5.8}$$

$$b_i = \sum_{g=1}^{G_i} n_{ig} M_{ig}^T \Sigma_{ig}^{-1} m_{ig} + \nu_i \Sigma_i^{-1} m_i. \tag{5.9}$$

5.2.2.2 Gaussian-inverse-Wishart means and covariances

Under this model, data points are generated from Gaussian distributions with random mean and covariance parameters. More precisely, the parameters associated with label i are distributed as $\mu_i|\Sigma_i \sim \mathrm{N}(m_i, \frac{1}{\nu_i}\Sigma_i)$ and $\Sigma_i \sim \mathrm{IW}(\kappa_i, \Psi_i)$, where the covariance has inverse-Wishart distribution

$$f(\Sigma_i) = \frac{|\Psi_i|^{\kappa_i/2}}{2^{\kappa_i d/2}\Gamma_d(\kappa_i/2)} |\Sigma_i|^{\frac{\kappa_i+d+1}{2}} \exp\big(-\frac{1}{2}\mathrm{tr}(\Psi_i \Sigma_i^{-1})\big). \tag{5.10}$$

To compute the posterior probability of labeling function (5.6), we first marginalize out the mean parameters μ_i in a similar fashion to (5.7), obtaining

$$\begin{aligned} P_S(\phi_S) \propto\ & P(\phi_S) \prod_{\substack{i=1\\ n_i\geq 1}}^{l} \int \Bigg[\prod_{g=1}^{G_i} |2\pi\Sigma_{ig}|^{-n_{ig}/2} \times \exp\{-\frac{1}{2}\mathrm{tr}\big(\Psi_{ig}(\Sigma_{ig})^{-1}\big)\} \times \\ & (\nu_i)^{d/2} |\Sigma_i|^{-1/2} |A_i/(2\pi)|^{-1/2} \times \\ & \exp\Big\{ -\frac{1}{2}\big[\sum_{g=1}^{G_i} n_{ig} m_{ig}^T \Sigma_{ig}^{-1} m_{ig} + \nu_i m_i^T \Sigma_i^{-1} m_i - b_i^T A_i^{-1} b_i\big]\Big\}\Bigg] f(\Sigma_i) d\Sigma_i. \end{aligned} \tag{5.11}$$

The integration in the above equation has no closed form solution, thus we resort to Monte

Carlo integration for approximating it. Specifically, denoting the term in the brackets in equation (5.11) as $g(\Sigma_i)$, we draw J samples $\Sigma_i^{(j)} \sim \text{IW}(\kappa_i, \Psi_i)$, $j = 1, 2, ..., J$, and then compute the integral as $\frac{1}{J}\sum_{j=1}^{J} g(\Sigma_i^{(j)})$.

5.3 Results and Discussion

The performance of the proposed method for optimal clustering with missing values at random is compared with some suboptimal versions, two other methods for clustering data with missing values, and also classical clustering algorithms with imputed missing values. The performance comparison is carried out on synthetic data generated from different Gaussian RLPP models with different missing probability setups, and also on a publicly available dataset of breast cancer generated by TCGA Research Network (https://cancergenome.nih.gov/). In our experiments, the results of the exact optimal solution for the RLPP with missing at random (Optimal) is provided for smaller point sets, i.e. wherever computationally feasible. We have also tested two suboptimal solutions, similar to the ideas in [132], for an RLPP with missing at random. In the first one (Subopt. Pmax), the set of reference partitions ($\mathcal{R}_S$) is restricted to a closed ball of a specified radius centered on the partition with the highest probability, where the distance of two partitions is defined as the minimum Hamming distance between labels inducing the partitions. In both Optimal and Pmax, the reference set is further constrained to the partitions that assign the correct number of points to each cluster, but the set of candidate partitions ($\mathcal{C}_S$) includes all the possible partitions of n points, i.e. 2^{n-1}. In the second suboptimal solution (Subopt. Pseed), a local search within Hamming distance at 1 is performed starting from five random initial partitions to approximately find the partition with (possibly local) maximum probability. Then the sets of reference and candidate partitions are constrained to the partitions with correct cluster sizes with a specified Hamming distance from the found (local) maximum probability partition. The bounds derived in [132] for reducing the set of candidate and reference partitions are used, where applicable, in Optimal, Pseed, and Pmax.

In all scenarios, k-POD and fuzzy c-means with optimal completion strategy (FCM-OCS) are directly applied to the data with missing values. In the simulations in [141], where FCM-OCS is

introduced, to initialize cluster centers, the authors apply ordinary fuzzy c-means to the complete data, i.e. using knowledge of the missing values. To have a fair comparison with other methods, we calculate the initial cluster centers for FCM-OCS by applying fuzzy c-means to the subset of points with no missing features for lower missing rates. For higher missing rates we impute the missing values by the mean of the corresponding feature values across all points, and then apply fuzzy c-means to all the points to initialize the cluster centers. In order to apply the classical algorithms, the missing values are imputed according to [145], by employing a multivariate Gibbs sampler that iteratively generates samples for missing values and parameters based on the observed data. The classical algorithms included in our experiments include *k*-means (KM), fuzzy *c*-means (FCM), hierarchical clustering with single linkage (Hier. (Si)), and hierarchical clustering with complete linkage (Hier. (Co)). Moreover, completely random clusterer (Random) results are also included for performance comparisons.

5.3.1 Simulated Data

In the simulation analysis, the number of clusters is fixed at 2, and the dimensionality of the RLPPs (number of features) is set to 5. Additional results for 20 features are available in Additional file 1 of [146]. Point generation is done based on a Gaussian mixture model (GMM). Three different scenarios for the parameters of the GMM are considered: *i*) Fixed known means and covariances *ii*) Known covariances and unknown means with Gaussian distributions. *iii*) Unknown means and covariances with Gaussian inverse-Wishart distributions. We select the values of the parameters of the point generation process to have an approximate Bayes error of 0.15. The selected values are shown in Table 5.1. For the point set generation, the number of points from each cluster is fixed *a priori*. The distributions are first drawn from the assumed model, and then the points are generated based on the drawn distributions. A subset of the points' features is randomly selected to be hidden based on missing at random with different missing probabilities. Four different setups for the number of points are considered in our simulation analysis: 10 points from each cluster ($n_1 = n_2 = 10$), 12 points from one cluster and 8 points from the other cluster ($n_1 = 12, n_2 = 8$), 35 points from each cluster ($n_1 = n_2 = 35$), and 42 points from one cluster and 28 points from the

Table 5.1: Parameters for the point generation under three models. N, IW, $\mathbf{1}_d$, and I_d denote Gaussian, inverse-Wishart, column vector of all ones with length d, and $d \times d$ idendity matrix, respectively.

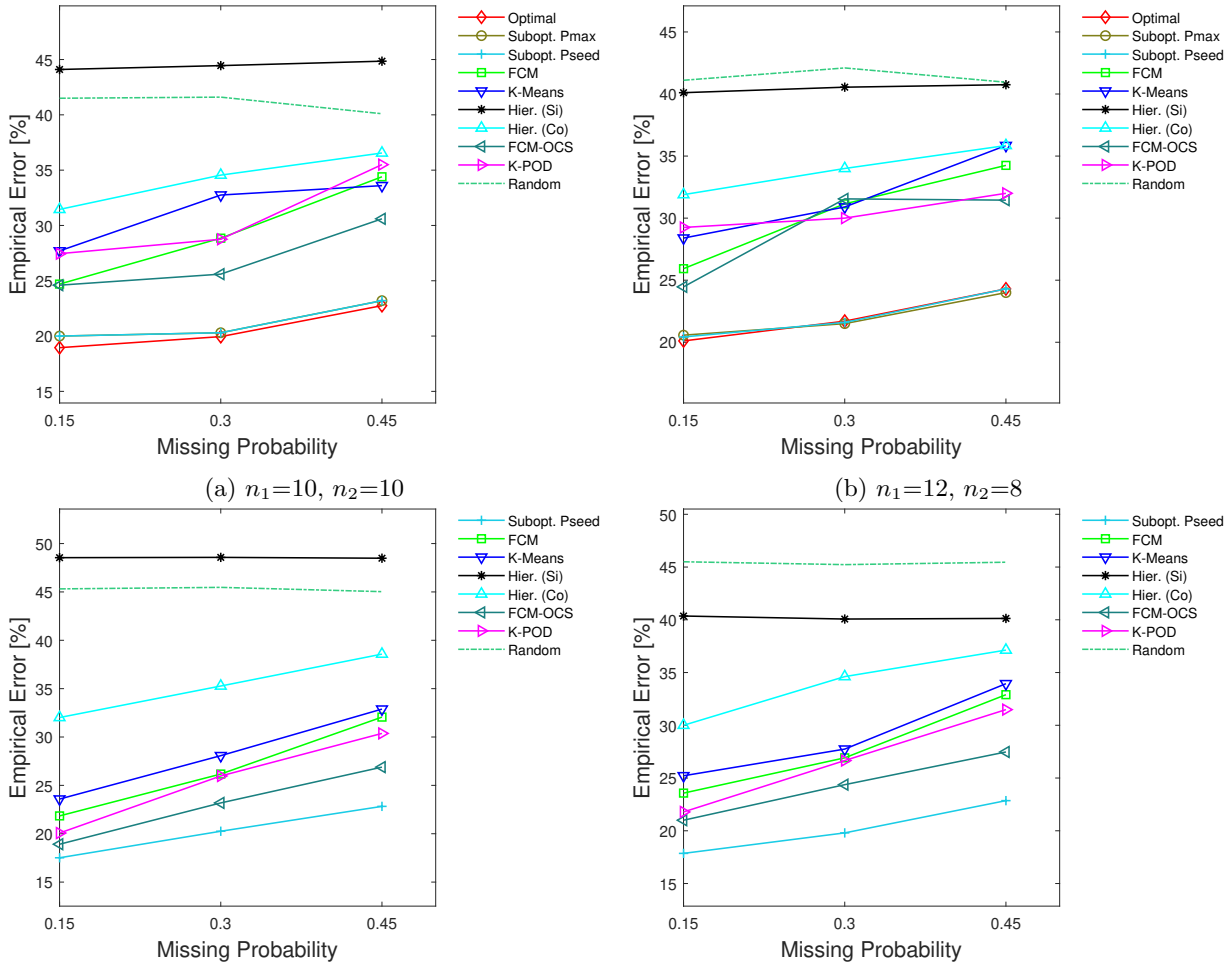
Model	Mean vectors	Covariance matrices	Distributions' hyperparameters
Fixed means and covariances	$\mu_1 = 0 \cdot \mathbf{1}_d, \mu_2 = 0.445 \cdot \mathbf{1}_d$	$\Sigma_1 = \Sigma_2 = 0.23 \cdot I_d$	—
Gaussian means and fixed covariances	$\mu_1 \sim \mathrm{N}(m_1, \frac{1}{\nu_1}\Sigma_1), \mu_2 \sim \mathrm{N}(m_2, \frac{1}{\nu_2}\Sigma_2)$	$\Sigma_1 = \Sigma_2 = 0.28 \cdot I_d$	$m_1 = 0 \cdot \mathbf{1}_d, m_2 = 0.45 \cdot \mathbf{1}_d,$ $\nu_1 = 30, \nu_2 = 5$
Gaussian means and inverse-Wishart covariances	$\mu_1 \sim \mathrm{N}(m_1, \frac{1}{\nu_1}\Sigma_1), \mu_2 \sim \mathrm{N}(m_2, \frac{1}{\nu_2}\Sigma_2)$	$\Sigma_1 \sim \mathrm{IW}(\kappa_1, \Psi_1)$,$\Sigma_2 \sim \mathrm{IW}(\kappa_2, \Psi_2)$	$m_1 = 0 \cdot \mathbf{1}_d, m_2 = 0.45 \cdot \mathbf{1}_d,$ $\nu_1 = 30, \nu_2 = 5,$ $\Psi_1 = \Psi_2 = 20.7 \cdot I_d,$ $\kappa_1 = \kappa_2 = 75$

other cluster ($n_1 = 42, n_2 = 28$). When having unequal sized clusters, in half of the repetitions n_1 points are generated from the first distribution and n_2 points from the second distribution, and vice-versa in the other half. In each simulation repetition, all clustering methods are applied to the points to generate a vector of labels that induces a two-cluster partition. The predicted label vector by each method is compared with the true label vector of each point in the point set to calculate the error of that method on that point set. In other words, for each method the number of points assigned to a cluster different from their true one are counted (after accounting for the label switching issue) and divided by the total number of points ($n = n_1 + n_2$) to calculate the clustering error of that method on the point set. These errors are averaged across all point sets in different repetitions to empirically estimate the clustering error of each method under a model and fixed missing-value probability. In cases with $n = 70$, since applying Optimal and Pmax is computationally prohibitive, we only provide the results for Pseed.

In Additional file 1 of [146], the average clustering errors are shown as a function of the Hamming distance threshold used to define the set of reference partitions in Pmax and Pseed, for different simulation scenarios. From the Figures in Additional file 1 of [146], we see that in all cases, the performances of Pmax and Pseed are quite insensitive to the set threshold of the Hamming distance for reference partitions. Note that in these types of figures all the other methods' performances other than Pmax and Pseed are constant in each plot.

The average results for the fixed mean vectors and covariance matrices across 100 repetitions

are shown in Figure 5.1. Here, the Hamming distance threshold for reference partitions in Pmax and Pseed is fixed at 1. It can be seen that Optimal, Pmax, and Pseed outperform all the other methods in all the smaller sample size settings, and Pmax and Pseed have virtually the same performance as Optimal. For the larger sample size settings where only Pseed is applied, its superior performance compared with other methods is clear from the figure.



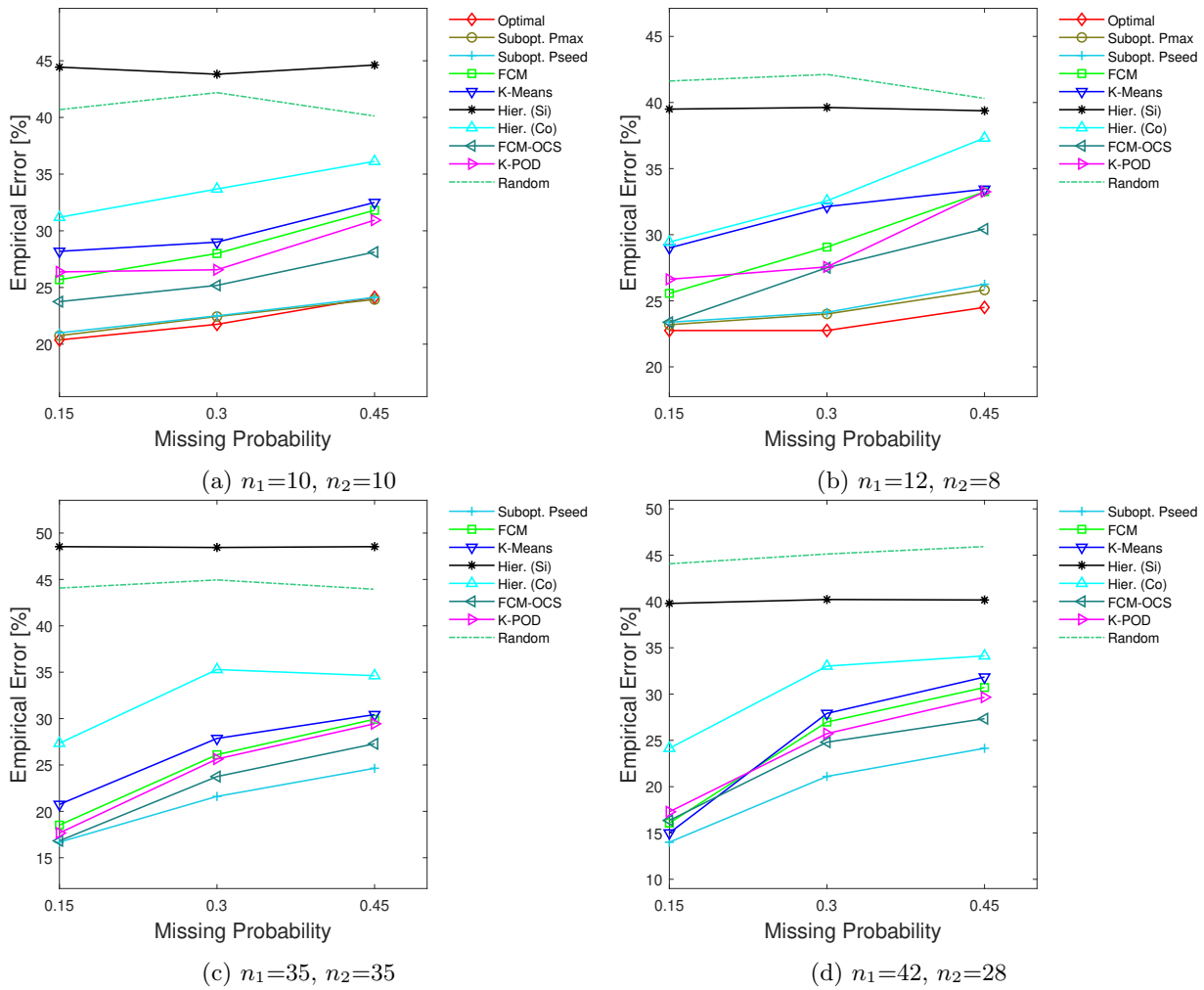
(a) n_1=10, n_2=10 (b) n_1=12, n_2=8

(c) n_1=35, n_2=35 (d) n_1=42, n_2=28

Figure 5.1: Average clustering errors vs. missing probability for fixed means and covariances model. The first and second rows correspond to $n = 20$ and $n = 70$, respectively.

Figure 5.2 depicts the comparison results under the unknown mean vectors with Gaussian distributions and fixed covariance matrices averaged over 80 repetitions. The Hamming distance

threshold in Pmax and Pseed is set to 2. For smaller sample sizes, Optimal, Pmax and Pseed have lower average errors than all the other methods. We can see that for balanced clusters the suboptimal and optimal solutions have very close performances, but for the unbalanced clusters case with higher missing probabilities the difference between Optimal and Pmax and Pseed gets noticeable. For larger sample sizes Pseed consistently outperforms the other methods, although for lower missing probabilities it has closer competitors. In all cases, as the missing probability increases, the superior performance of the proposed methods becomes more significant.

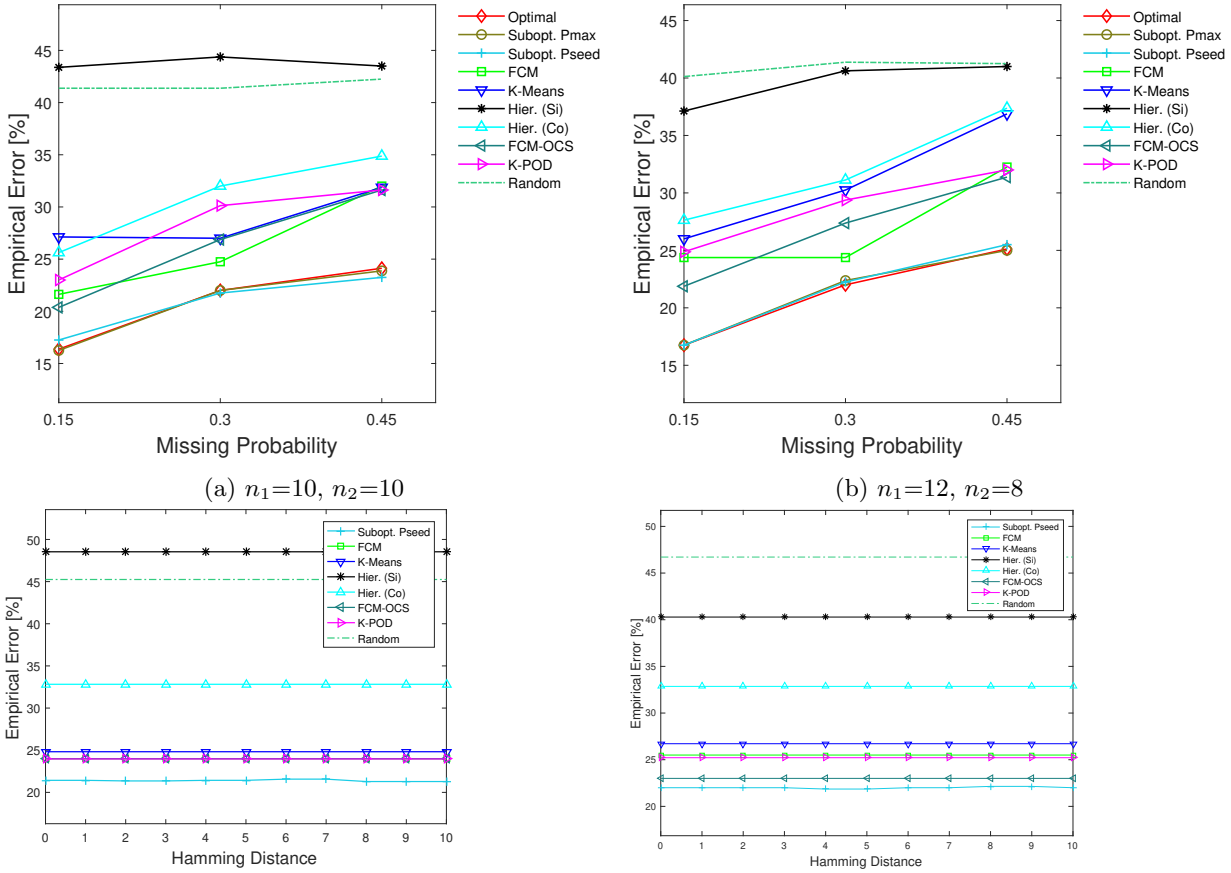


(a) n_1=10, n_2=10 (b) n_1=12, n_2=8

(c) n_1=35, n_2=35 (d) n_1=42, n_2=28

Figure 5.2: Average clustering errors vs. missing probability for Gaussian means and fixed covariances model. The first and second rows correspond to $n = 20$ and $n = 70$, respectively.

The average results under the unknown mean vectors and coavriance matrices with Gaussian-inverse-Wishart distribution over 40 repetitions are provided in Figure 5.3. In the smaller sample size cases, the Hamming distance threshold in Pmax and Pseed is fixed at 8, and we can see that the proposed suboptimal (Pmax and Pseed) and optimal clustering with missing values have very close average errors, and all are much lower than the other methods' errors. For larger sample sizes, only the results for missing probability equal to 0.15 are shown vs. the Hamming distance threshold used to define the reference partitions in Pseed. Again, Pseed performs better than the other methods.



(a) n_1=10, n_2=10

(b) n_1=12, n_2=8

(c) n_1=35, n_2=35, miss. prob.=0.15

(d) n_1=42, n_2=28, miss. prob.=0.15

Figure 5.3: Average clustering errors for Gaussian means and inverse-Wishart covariances model. The first row corresponds to $n = 20$, and the errors are shown for different missing probabilities. The second row corresponds to $n = 70$ and missing probability of 0.15, where the errors are plotted vs. the Hamming distance threshold used to define the reference partitions in Pseed.

5.3.2 RNA-seq Data

In this section the performance of the clustering methods are examined on a publicly available RNA-seq dataset of breast cancer. The data is available on The Cancer Genome Atlas (TCGA) [147], and is procured by the R package TCGS2STAT [148]. It consists of matched tumor and normal samples, and includes 97 points from each. The original data are in terms of the number of sequence reads mapped to each gene. RNA-seq data are integers, highly skewed and over-dispersed [113]. Thus, we apply a variance stabilizing transformation (VST) [149] implemented in DESeq2 package [150], and transform data to a log2 scale that have been normalized with respect to library size. For all subsequent analysis, other than for calculating clustering errors, we assume that the labels of data are unknown. Feature selection is performed in a completely unsupervised manner, since in clustering no labels are known in practice. The top 10 genes in terms of variance to mean ratio of expression are picked as features to be used in clustering algorithms. In general, for setting prior hyperparameters, external sources of information like signaling pathways, where available, can be leveraged [89, 88]. Here, we only use a subset of the discarded gene expressions, i.e. the next 90 top genes (in terms of variance to mean ratio of expression), for prior hyperparameters calibration for the optimal/suboptimal approaches. We follow the approach in [151] and employ the method of moments for prior calibration, but unlike [151], a single set of hyperparameters is estimated and used for both clusters, since the labels of data are not available. It is well known that in small sample size settings, estimation of covariance matrices, scatter matrices and even mean vectors may be problematic. Therefore, similar to [151], we assume the following structure

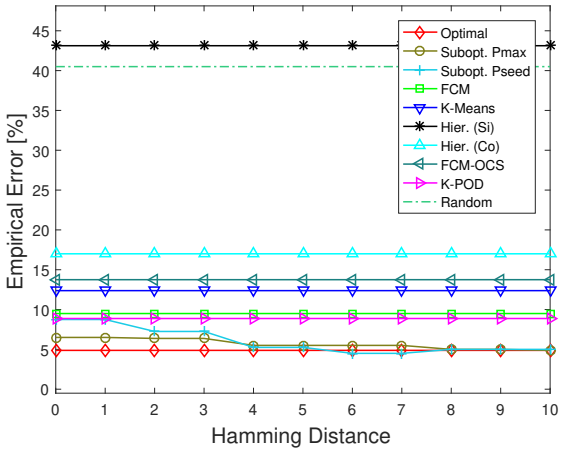
$$\Psi_0 = \Psi_1 = \begin{bmatrix} \sigma^2 & \rho\sigma^2 & \dots & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 & \dots & \rho\sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ \rho\sigma^2 & \dots & \dots & \sigma^2 \end{bmatrix}_{d\times d},$$

$$m_0 = m_1 = m[1, \cdots, 1]_d^T,$$

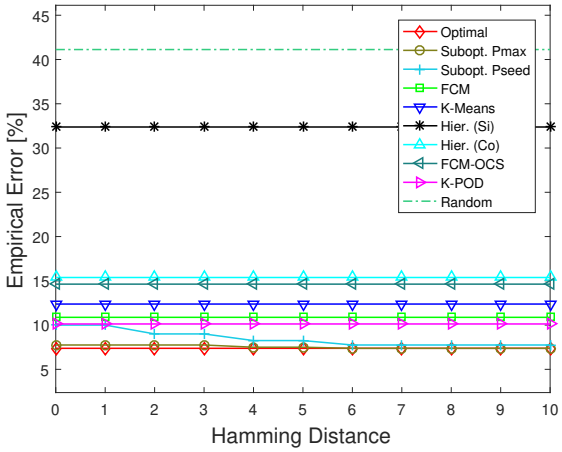
$$\nu_0 = \nu_1 = \nu, \kappa_0 = \kappa_1 = \kappa,$$

and estimate five scalars (m, σ^2, ρ, κ, ν) from the data.

In each repetition, stratified sampling is done, i.e. n_1 and n_2 points are sampled randomly from each group (normal and tumor). When $n_1 \neq n_2$, in half of the repetitions n_1 and n_2 points are randomly selected from the normal and tumor samples, respectively, and vice-versa in the other half. Prior calibration is performed in each repetition, and 15% of the selected features are considered as missing values. Similar to the experiments on the simulated data, the clustering error of each method in each iteration is calculated by comparing the predicted labels and true labels of the sampled points (accounting for label switching issue), and the average results over 40 repetitions are provided in Figure 5.4. It can be seen that the proposed optimal clustering with missing values and its suboptimal versions outperform the other algorithms. It is worth noting that the performance of Pseed is more sensitive to the selected Hamming distance threshold for reference partitions compared with the results on simulated data.



(a) n_1=10,n_2=10,miss. prob.=0.15



(b) n_1=12,n_2=8,miss. prob.=0.15

Figure 5.4: Empirical clustering errors on breast cancer RNA-seq data.

6. EXPERIMENT DESIGN UNDER MODEL UNCERTAINTY *

6.1 Introduction

Optimal experimental design is critical for autonomously learning physical models. This is because experiments can be costly and time-consuming, such as the ones in biology and materials design. It is desirable to help design the experiments that reduce the uncertainty pertaining to the ultimate operational objective, be it control, filtering, classification, drug design, materials design, or some other operational goal.

In the first part of this Chapter, we provide a generalized mean objective cost of uncertainty (MOCU) and the corresponding experimental design. MOCU quantifies the performance cost of using an operator that is optimal across an uncertainty class of systems as opposed to using an operator that is optimal for a particular system. MOCU-based experimental design selects an experiment to maximally reduce MOCU, thereby gaining the greatest reduction of uncertainty impacting the operational objective. We then show that the classical Knowledge Gradient and Efficient Global Optimization procedures are specific implementations of MOCU-based experimental design under their modeling assumptions.

In the second part of the Chapter, we develop an efficient experiment design framework for materials discovery accounting for model uncertainty. The accelerated exploration of the Materials Design Space (MDS) has been recognized for more than a decade as a key enabler for potentially transformative technological developments [152, 153]. The proposed method leverages prior knowledge in terms of potential models/feature sets where it adaptively learns the most promising regions in the materials space wile identifying the models that most efficiently guide such exploration.

*Parts of this Chapter are reprinted with permission from S. Boluki, X. Qian, and E. R. Dougherty "Experimental design via generalized mean objective cost of uncertainty." IEEE Access, vol. 7, 2223–2230, 2018. Copyright 2018 IEEE.
Parts of this Chapter are reprinted with permission from A. Talapatra*, S. Boluki*, T. Duong, X. Qian, and E. R. Dougherty, R. Arróyave "Autonomous efficient experiment design for materials discovery with Bayesian model averaging." Physical Review Materials, vol. 2, no. 11, 113803, 2018. Copyright 2018 APS. *:Equal contribution

6.2 Generalized Mean Objective Cost of Uncertainty

From the Bayesian perspective, Lindley's paradigm posits a general framework for Bayesian experimental design [154]. Two standard procedures within this paradigm are the Knowledge Gradient (KG) [155, 156] and Efficient Global Optimization (EGO) [157], which provide (one-step) optimal experimental design under Gaussian belief and observation noise (KG only) for an offline ranking and selection problem. A more recently introduced method is based on the mean objective cost of uncertainty (MOCU), which quantifies the performance cost of using an operator that is optimal across an uncertainty class of systems as opposed to an operator that is optimal for a particular system within the class [158]. MOCU-based experimental design selects an experiment that maximally reduces MOCU, thereby optimally reducing uncertainty with respect to the operational objective [159].

Here we consider a generalized formulation of MOCU that is neither necessarily dependent on the particularities of the underlying system model nor necessarily involves a design problem focused on operators. In [129] we show that the corresponding generalized experimental design encompasses existing formulations in signal processing, genomics, and materials discovery. Here, we show that it fits within Lindley's paradigm for Bayesian experimental design. Within this generalized framework we examine the connection and differences of MOCU-based formulations with other Bayesian experimental design methods. In particular, we show that the generalized MOCU generates the same policies as Knowledge Gradient and Efficient Global Optimization under their modeling assumptions. Not only does the generalized MOCU framework unify disparate problems, it opens up Bayesian experimental design for reduction of objective related uncertainty.

6.2.1 Generalized MOCU

We first formulate experimental design in terms of generalized MOCU. In this section, the lower case Greek letters denote random variables or distribution functions and capital Greek letters denote the corresponding domain space. We assume a probability space Θ with probability measure π, a set Ψ, and a function $C : \Theta \times \Psi \to [0, \infty)$, where Θ, π, Ψ, and C are called the *un-*

certainty class, *prior distribution*, *action space*, and *cost function*, respectively. Elements of Θ and Ψ are called *uncertainty parameters* and *actions*, respectively. For any $\theta \in \Theta$, an *optimal action* is an element $\psi_\theta \in \Psi$ such that $C(\theta, \psi_\theta) \leq C(\theta, \psi)$ for any $\psi \in \Psi$. An *intrinsically Bayesian robust (IBR) action* is an element $\psi^{\Theta}_{\mathrm{IBR}} \in \Psi$ such that $\mathrm{E}_\theta[C(\theta, \psi^{\Theta}_{\mathrm{IBR}})] \leq \mathrm{E}_\theta[C(\theta, \psi)]$ for any $\psi \in \Psi$.

Whereas $\psi^{\Theta}_{\mathrm{IBR}}$ is optimal over Θ, for $\theta \in \Theta$, ψ_θ is optimal relative to θ. The *objective cost of uncertainty* is defined by the performance loss of applying $\psi^{\Theta}_{\mathrm{IBR}}$ instead of ψ_θ on θ:

$$\mathrm{U}_\Psi(\Theta) = C(\theta, \psi^{\Theta}_{\mathrm{IBR}}) - C(\theta, \psi_\theta). \tag{6.1}$$

Averaging this cost over Θ gives the *mean objective cost of uncertainty (MOCU)*:

$$\mathrm{M}_\Psi(\Theta) = \mathrm{E}_\theta[C(\theta, \psi^{\Theta}_{\mathrm{IBR}}) - C(\theta, \psi_\theta)]. \tag{6.2}$$

The action space is arbitrary so long as the cost function is defined on $\Theta \times \Psi$. It can be a set of filters defined on a random process with C being mean-square error or a set of drug interventions with C quantifying patient condition.

In decision theory, *regret* is defined as the difference between the maximum payoff (for making an optimal decision) and the actual payoff (for the decision that has been made). From this perspective, MOCU can be viewed as the minimum expected regret for using a robust operator.

Suppose there is a set Ξ, called the *experiment space*, whose elements, ξ, called *experiments*, are jointly distributed with the uncertainty parameters θ. To avoid overly complex notation, we denote both an experiment and its outcome by ξ. More specifically, when used in conditioning the probability spaces and distributions, ξ represents an outcome, and when in a minimization/maximization argument, it corresponds to an experiment. Given $\xi \in \Xi$, the conditional distribution $\pi(\theta|\xi)$ is the *posterior distribution* relative to ξ and $\Theta|\xi$ denotes the corresponding probability space, called the *conditional uncertainty class*. Relative to $\Theta|\xi$, we define IBR actions

$\psi_{\mathrm{IBR}}^{\Theta|\xi}$ and the conditional (remaining) MOCU,

$$\mathrm{M}_{\Psi}(\Theta|\xi) = \mathrm{E}_{\theta|\xi}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta|\xi}) - C(\theta, \psi_{\theta})], \tag{6.3}$$

where the expectation is with respect to $\pi(\theta|\xi)$. Taking the expectation over ξ gives the expected remaining MOCU,

$$\mathrm{D}_{\Psi}(\Theta, \xi) = \mathrm{E}_{\xi}[\mathrm{M}_{\Psi}(\Theta|\xi)] = \mathrm{E}_{\xi}[\mathrm{E}_{\theta|\xi}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta|\xi}) - C(\theta, \psi_{\theta})]], \tag{6.4}$$

which is called the *experimental design value*. An optimal experiment $\xi^* \in \Xi$ minimizes $\mathrm{D}_{\Psi}(\Theta, \xi)$, i.e.,

$$\xi^* = \operatorname*{argmin}_{\xi \in \Xi} \mathrm{D}_{\Psi}(\Theta, \xi). \tag{6.5}$$

ξ^* also minimizes the difference between the expected remaining MOCU and the current MOCU (maximizes the absolute difference):

$$\begin{aligned} \xi^* &= \operatorname*{argmin}_{\xi \in \Xi} \mathrm{D}_{\Psi}(\Theta, \xi) - \mathrm{M}_{\Psi}(\Theta) = \\ &\quad \operatorname*{argmin}_{\xi \in \Xi} \mathrm{E}_{\xi}[\mathrm{E}_{\theta|\xi}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta|\xi}) - C(\theta, \psi_{\theta})]] - \mathrm{E}_{\theta}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta}) - C(\theta, \psi_{\theta})] \\ &= \operatorname*{argmin}_{\xi \in \Xi} \mathrm{E}_{\xi}[\mathrm{E}_{\theta|\xi}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta|\xi})]] - \mathrm{E}_{\theta}[C(\theta, \psi_{\mathrm{IBR}}^{\Theta})]. \end{aligned} \tag{6.6}$$

There is wide flexibility in experimental design, depending on the assumptions regarding the uncertainty class, action space, and experiment space, leading to many existing Bayesian experimental design formulations. Bayesian experimental design has a long history, in particular, utilizing the expected gain in Shannon information [160, 161, 162, 163]. In 1972, Lindley proposed a general decision theoretic approach incorporating a two-part decision involving the selection of an experiment followed by a terminal decision [154]. Supposing λ is a design selected from a family

Λ and $\mathbf{X}$ is a data vector, and leaving out the terminal decision, an optimal experiment is given by

$$\lambda^* = \arg\max_{\lambda \in \Lambda} \mathrm{E}_{\mathbf{X}}[\mathrm{E}_{\Theta}\left[U(\theta, \mathbf{X}, \lambda)|\mathbf{X}, \lambda\right]|\lambda], \tag{6.7}$$

where U is a utility function (see [164] for the full decision-theoretic optimization).

With generalized MOCU, recalling that ξ represents both an experiment and its outcome, each experiment ξ corresponds to a data vector $\mathbf{X}|\xi$ and the expected remaining MOCU is

$$\mathrm{E}_{\xi}[\mathrm{M}_{\Psi}(\Theta|\mathbf{X}, \xi)] = \mathrm{E}_{\mathbf{X}|\xi}[\mathrm{E}_{\Theta}[C_{\theta|(\mathbf{X}|\xi)}(\psi_{\mathrm{IBR}}^{\Theta|(\mathbf{X}|\xi)}) - C_{\theta|(\mathbf{X}|\xi)}(\psi_{\theta|(\mathbf{X}|\xi)})]] = \mathrm{E}_{\mathbf{X}|\xi}[\mathrm{E}_{\Theta}[\mathrm{U}_{\Psi}(\theta, \mathbf{X}, \xi; \Theta)]]. \tag{6.8}$$

From (6.8), the optimization of (6.5) can be expressed in the same form as (6.7), with ξ in place of λ and utility function $-\mathrm{U}_{\Psi}(\theta, \mathbf{X}, \xi; \Theta)$.

Hence, in descending order of generality, we have Lindley's procedure, generalized MOCU, and MOCU.

With sequential experiments, the action space and experiment space can be time dependent, i.e., they can be different for each time step. Hereafter, in sequential experiment setups, the action space and experiment space at time step t, and the optimal experiment selected at t to be performed at the next time step are denoted by Ψ^t, Ξ^t, and $\xi^{*,t}$, respectively. Let $\pi(\theta|\xi^{:t})$ be the posterior distribution given the selected experiments' outcomes from the first time step through t, and $\Theta|\xi^{:t}$ denote the corresponding conditional uncertainty class. When experiments are selected sequentially and there is no fixed limited budget of experiments but instead the experimenter wants to stop the iterative procedure when only negligible knowledge regarding the objective can be gained from additional experiments, the form in (6.6) is useful because it incorporates the difference between the expected remaining MOCU and the current MOCU.

Sequential experiments can be understood in terms of a design loop for designing optimal operators under uncertainty. Referring to Figure 6.1, and considering the standard MOCU formulation, the base of the design loop is construction of the prior. This can be done in numerous ways; how-

ever, a very general procedure can be used to derive the *Maximal Knowledge-driven Information Prior (MKDIP)* (Chapters 2 and 3) that minimizes an information-based cost function subject to constraints characterizing our prior knowledge. The prior can then be updated to a posterior using data. Assuming the existence of effective characteristics, following posterior update, these are computed and an IBR operator determined. Uncertainty is quantified by MOCU and, if desired, optimal experiments performed to produce new knowledge that can be used to supplement the original knowledge or directly condition the original prior, in either case producing a new prior to re-institute the design process. The design loop involves two optimizations, and therefore two cost functions, one for prior construction and one for operator design.

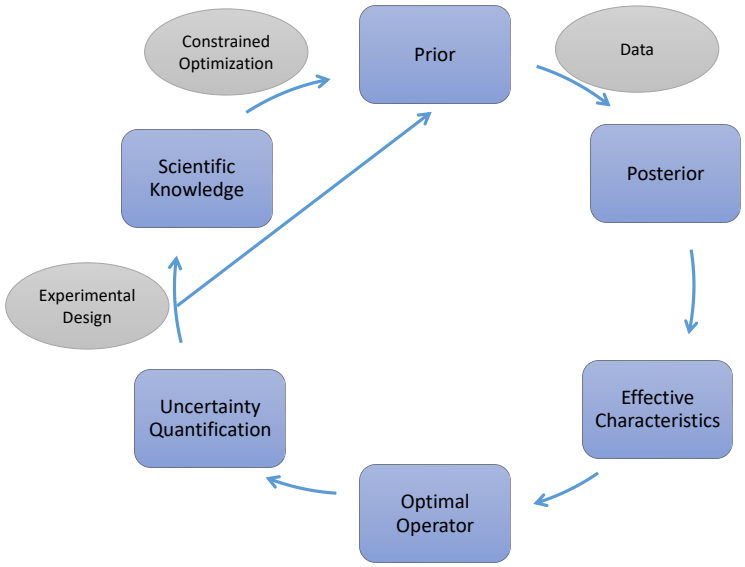


Figure 6.1: A design loop for designing optimal operators under uncertainty.

In generalized MOCU, the parameters of the cost function can come from an underlying physical system. Another possibility is that they correspond to the surrogate model, instead of the actual physical model, which is used for the experimental design. A third possibility is that we do not possess a physical model and we lack sufficient knowledge to posit a surrogate model relating to our objective. Nevertheless, we can take an ad hoc approach and select a model with known predictive

properties. This model can be kernel-based model, for instance, a Gaussian Process Regression model [165]. More generally, the model can consist of a set of possible parametric families, or be a kernel-based model with different possible feature sets, or even kernel-based models with different choices for the kernel function. In [166] and Section 6.3 no knowledge is assumed regarding which feature set or model family would be the best. Instead, Bayesian model averaging is used and models are weighted by their posterior probabilities of being the correct model, where possible models or feature sets are selected based on domain knowledge. Assuming a single objective, generalized MOCU can be applied to all three scenarios.

6.2.2 Connection of MOCU-based Experimental Design with KG and EGO

Knowledge Gradient (KG) [155, 156], which is used in different fields, from drug discovery to material design [167, 168], was originally introduced as a solution to an offline ranking and selection problem, where the assumption is that there are $A \geq 2$ actions (alternatives) that can be selected, i.e., $\Psi = \{\psi_1, \ldots, \psi_A\}$. Each action has an unknown true reward (sign-flipped cost) and at each time step an experiment provides a noisy observation of the reward of a selected action. There is a limited budget (B) of the number of measurements we can make before the time arrives to decide which action is the best, that being the one having the lowest expected cost (or the highest expected reward).

The assumption is that we have Gaussian prior beliefs over the unknown rewards, either independent Gaussian beliefs over the rewards when the rewards of different actions are uncorrelated, or a joint Gaussian belief when the rewards are correlated. In the independent case, for each action-reward pair $(\psi_i, \theta_{\psi_i})$, $\theta_{\psi_i} \sim N(m_{\psi_i}, \beta_{\psi_i})$. In the correlated case, the vector of rewards, $[\theta_{\psi_1}, \ldots, \theta_{\psi_A}]$, has a multivariate Gaussian distribution $N(m, \Sigma)$ with the mean vector $m = [m_{\psi_1}, \ldots, m_{\psi_A}]$ and covariance matrix Σ, with diagonal entries $[\beta_{\psi_1}, \ldots, \beta_{\psi_A}]$. If the selected action to be applied at t is ψ^t, then the observed noisy reward of ψ^t at that iteration is $\xi^t = \theta_{\psi^t} + \epsilon^t$, where θ_{ψ^t} is unknown and $\epsilon^t \sim N(0, \lambda_{\psi^t})$ is independent of the reward of ψ^t.

Here, the underlying system to learn is the unknown reward function and each possible model is fully described by a reward vector $\theta = [\theta_{\psi_1}, \theta_{\psi_2}, \ldots, \theta_{\psi_A}]$ in the uncertainty class Θ. For the

independent case, $\pi(\theta) = \prod_{i=1}^{A} N(m_{\psi_i}, \beta_{\psi_i})$. For the correlated case, $\pi(\theta) = N(m, \Sigma)$. The experiment space is $\Xi = \{\xi_1, \ldots, \xi_A\}$, where experiment ξ_i corresponds to applying ψ_i and getting a noisy observation of its reward θ_{ψ_i}, that is, measuring θ_{ψ_i} with observation noise, where $\xi_i|\theta_{\psi_i} \sim N(\theta_{\psi_i}, \lambda_{\psi_i})$. In the independent case the state of knowledge at each time point t is captured by the posterior values of the means and variances for the rewards after incorporating observations $\xi^{:t}$ as $S^t = [(m^t_\psi, \beta^t_\psi)]_{\psi\in\Psi}$, and in the correlated case by the posterior vector of means and a covariance matrix after observing $\xi^{:t}$ as $S^t = (m^t, \Sigma^t)$, where $m^t = [m^t_{\psi_1}, \ldots, m^t_{\psi_A}]$ and the diagonal of Σ^t is the vector $[\beta^t_{\psi_1}, \ldots, \beta^t_{\psi_A}]$. The probability space $\Theta|\xi^{:t}$ is equal to $\Theta|S^t$ and the cost function is $C(\theta, \psi) = -\theta_\psi$.

For this problem, the IBR action at time step t is

$$\psi^{\Theta|\xi^{:t}}_{\text{IBR}} = \operatorname*{argmin}_{\psi\in\Psi} \mathrm{E}_{\Theta|\xi^{:t}}\big[C(\theta, \psi)\big] = \operatorname*{argmin}_{\psi\in\Psi} \mathrm{E}_{\Theta|\xi^{:t}}\big[-\theta_\psi\big] = \operatorname*{argmax}_{\psi\in\Psi} \mathrm{E}_{\Theta|\xi^{:t}}\big[\theta_\psi\big] = \operatorname*{argmax}_{\psi\in\Psi} m^t_\psi. \tag{6.9}$$

By (6.4) and (6.5), the optimal experiment selected at time step t (to be performed at $t+1$) can be derived:

$$\begin{aligned}
\xi^{*,t} &= \operatorname*{argmin}_{\xi_i\in\Xi} \mathrm{E}_{\xi_i|\xi^{:t}}[\mathrm{E}_{\theta|\xi_i,\xi^{:t}}[C(\theta, \psi^{\Theta|\xi^{:t},\xi_i}_{\text{IBR}})]] - \mathrm{E}_{\theta|\xi^{:t}}[C(\theta, \psi^{\Theta|\xi^{:t}}_{\text{IBR}})] \\
&= \operatorname*{argmin}_{\xi_i\in\Xi} \mathrm{E}_{\xi_i|\xi^{:t}}\Big[\mathrm{E}_{\theta|\xi^{:t+1}}\big[-\theta_{\psi^{\Theta|\xi^{:t+1}}_{\text{IBR}}}\big]\Big] - \mathrm{E}_{\theta|\xi^{:t}}\big[-\theta_{\psi^{\Theta|\xi^{:t}}_{\text{IBR}}}\big] \\
&= \operatorname*{argmax}_{\xi_i\in\Xi} \mathrm{E}_{\xi_i|\xi^{:t}}\Big[\mathrm{E}_{\theta|\xi^{:t+1}}\big[\theta_{\psi^{\Theta|\xi^{:t+1}}_{\text{IBR}}}\big]\Big] - \mathrm{E}_{\theta|\xi^{:t}}\big[\theta_{\psi^{\Theta|\xi^{:t}}_{\text{IBR}}}\big] \\
&= \operatorname*{argmax}_{\xi_i\in\Xi} \mathrm{E}_{\xi_i|\xi^{:t}}\Big[\max_{\psi'\in\Psi} m^{t+1}_{\psi'}\Big] - \max_{\psi'\in\Psi} m^t_{\psi'}.
\end{aligned} \tag{6.10}$$

The policy (6.10) derived by direct application of the generalized MOCU is exactly the same as the original KG policy in [155], [156], and [169]. As KG is shown to be optimal when the horizon is a single measurement and asymptotically optimal (the number of measurements goes to infinity), the same holds for the MOCU-based policy for this problem.

Efficient Global Optimization (EGO) [157], which is based on expected improvement (EI), is

widely used for black-box optimization and experimental design. As shown in [168], KG reduces to EGO when there is no observation noise and choosing the best action at each time step is limited to selecting from the set of actions whose rewards have been previously observed; that is, at each time step if we want to make a final decision as to the best action to be applied, it must be an action whose performance has been previously observed from the first time step up to that time. Thus, MOCU-based learning can also be reduced to EGO under its model assumptions.

6.3 Efficient Experiment Design for Materials Discovery

The accelerated exploration of the materials space in order to identify configurations with optimal properties is an ongoing challenge. Current paradigms are typically centered around the idea of performing this exploration through high-throughput experimentation/computation. Such approaches, however, do not account for|the always present|constraints in resources available. Recently, this problem has been addressed by framing materials discovery as an optimal experiment design [170]. This work augments earlier efforts by putting forward a framework that efficiently explores the materials design space not only accounting for resource constraints but also incorporating the notion of model uncertainty. The resulting approach combines Bayesian Model Averaging within Bayesian Optimization in order to realize a system capable of autonomously and adaptively learning not only the most promising regions in the materials space but also the models that most efficiently guide such exploration.

6.3.1 Bayesian Optimization under Model Uncertainty

Small sample sizes are ubiquitous in materials science. Experiments—and simulations—are often resource-intensive and this imposes significant constraints on any attempt to explore/exploit the MDS. Moreover, in the absence of sufficient information, there are, *a priori*, multiple features that are potentially predictive of the material performance metric of interest. In all the well-known experiment design methods in the literature, one must select the model (the set of predictive features and/or the parametric form or the kernel functional form of the model) before starting the experiment design loop.

Unfortunately, due to small sample size and large number of potential predictive models, the model selection step may not result in the true best predictive model for efficient Bayesian Optimization [171, 172]. It has been shown that small sample sizes pose a great challenge in model selection due to inherent risk of imprecision and overfitting [171, 172], and no feature selection method performs well in all scenarios when sample sizes are small [173]. Thus, by selecting a single model as the predictive model based on small observed sample data, one ignores the model uncertainty [174].

6.3.2 Building Robust Predictive Models through Bayesian Model Averaging

One possible approach to circumvent this problem is to weight all the possible models by their corresponding probability of being the true model, and use all of these in the experiment design step so that model uncertainty can be taken care of for Bayesian Optimization. In other words, the derived predictive model is a marginalized aggregation of all the potential predictive models, weighted by the prior probability and likelihood of the observed data for that model, resulting in the Bayesian Model Averaging (BMA) method [175, 176].

Here, we discuss the multi-output case from which the single output can be readily deduced. Let y^j represent the jth output of interest, and $\boldsymbol{x}$ the corresponding vector of features or materials design parameters, and the observed data be denoted by $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\}$, where $\boldsymbol{Y} = [Y^1, ..., Y^q]$ is a matrix having the collection of the observed j^{th} output as its jth column, i.e. $Y^j = [y_1^j, ..., y_n^j]^T$, where n is the number of observed data points, and $\boldsymbol{X}$ represent the matrix of the collection of the corresponding observed features. Here, to simplify the notation we have dropped the subscript denoting the experiment iteration step for $\mathcal{D}$, but note that $\mathcal{D} = \mathcal{D}_{\text{n}}$ at any nth step. The predictive probabilistic model for $\boldsymbol{y}$ for a new feature vector $\boldsymbol{x}$ after observing $\mathcal{D}$ is

$$P(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}) = \sum_{i=1}^{L} P(M_i|\mathcal{D}) P(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}, M_i), \tag{6.11}$$

where $P(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}, M_i)$ represents each potential probabilistic predictive model, and

$$P(M_i|\mathcal{D}) = \frac{P(\mathcal{D}|M_i)P(M_i)}{\sum_{j=1}^{L} P(\mathcal{D}|M_j)P(M_j)}, \tag{6.12}$$

$$P(\mathcal{D}|M_i) = \int P(\mathcal{D}|\boldsymbol{\theta}_i, M_i)P(\boldsymbol{\theta}_i|M_i)d\boldsymbol{\theta}_i, \tag{6.13}$$

are the (posterior) probability of each model being the true predictive model, and the marginal probability of the observed data under model M_i, respectively. L is the total number of models under consideration, and M_i and $\boldsymbol{\theta}_i$ represents the ith model and the vector of ith model parameters, respectively.

If we further assume independence among outputs and let $\mathcal{D}_j$ denote $\{\boldsymbol{X}, Y^j\}$, we have $P(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D}, M_i) = \prod_{j=1}^{q} P(y^j|\boldsymbol{x}, \mathcal{D}_j, M_i)$ and

$$P(\mathcal{D}|M_i) = \prod_{j=1}^{q} P(\mathcal{D}_j|M_i) = \prod_{j=1}^{q} \int P(\mathcal{D}_j|\boldsymbol{\theta}_i^j, M_i)P(\boldsymbol{\theta}_i^j|M_i)d\boldsymbol{\theta}_i^j. \tag{6.14}$$

When each potential probabilistic predictive model M_i is a Gaussian Process Regression (GPR) model [165], $\boldsymbol{\theta}_i^j$ are the parameters of the covariance function. In fact, each GPR model M_i is defined by a mean (basis) function ($m_i^j(\cdot)$) and a covariance function ($K_i^j(\cdot, \cdot; \boldsymbol{\theta}_i^j)$). In this setup, $P(y^j|\boldsymbol{x}, \mathcal{D}, M_i)$ is a Gaussian distribution, i.e. $P(y^j|\boldsymbol{x}, \mathcal{D}, M_i) = \mathcal{N}(\mu_i^j(\boldsymbol{x}), \sigma_i^{2,j}(\boldsymbol{x}))$, where the predicted mean and variance of the jth objective function are [165]:

$$\begin{aligned}
\mu_i^j(\boldsymbol{x}) &= m_i^j(\boldsymbol{x}) + K_i^j(\boldsymbol{x}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)K_i^j(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)^{-1}(Y^j - m_i^j(\boldsymbol{X})), \\
\sigma_i^{2,j}(\boldsymbol{x}) &= K_i^j(\boldsymbol{x}, \boldsymbol{x}; \boldsymbol{\theta}_i^j) - K_i^j(\boldsymbol{x}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)K_i^j(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)^{-1}K_i^j(\boldsymbol{X}, \boldsymbol{x}; \boldsymbol{\theta}_i^j).
\end{aligned} \tag{6.15}$$

In practice, when using type II maximum likelihood (ML-II) estimation, the covariance function parameters of each model are estimated by maximizing the marginal log-likelihood of the

observed data under that model, i.e. an estimate $\hat{\boldsymbol{\theta}}_i^j$ is calculated by maximizing

$$\log P(D_j|\boldsymbol{\theta}_i^j, M_i) = -\tfrac{1}{2}(Y^j - m_i^j(\boldsymbol{X}))^T K_i^j(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)^{-1}(Y^j - m_i^j(\boldsymbol{X})) - \tfrac{1}{2}|K_i^j(\boldsymbol{X}, \boldsymbol{X}; \boldsymbol{\theta}_i^j)| - \tfrac{n}{2}\log 2\pi, \tag{6.16}$$

where $|\cdot|$ denotes matrix determinant. A quasi-Newton method with multiple random starts can be employed to find the maximum of (6.16). This estimate $\hat{\boldsymbol{\theta}}_i^j$ is then used in (6.15) for prediction purposes under the model assumptions.

For a GPR, $P(\mathcal{D}_j|\boldsymbol{\theta}_i^j, M_i)$ is a multivariate Gaussian probability density function, and $P(\mathcal{D}_j|M_i) = \int P(\mathcal{D}_j|\boldsymbol{\theta}_i^j, M_i)P(\boldsymbol{\theta}_i^j|M_i)d\boldsymbol{\theta}_i^j$, the marginal probability of the observed data corresponding to j^{th} output under model M_i in (6.13), can be approximated by either first-order expansion of the exponent, or second-order expansion of the exponent known as Laplace approximation method [165]. In the first-order approximation, since $\hat{\boldsymbol{\theta}}_i^j$ is a stationary point of (6.16), $P(\mathcal{D}_j|M_i)$ can be approximated by $P(\mathcal{D}_j|\hat{\boldsymbol{\theta}}_i^j, M_i)$. In the second-order approximation, $P(\mathcal{D}_j|M_i) \approx P(\mathcal{D}_j|\hat{\boldsymbol{\theta}}_i^j, M_i) \int \exp\big(-\tfrac{1}{2}(\boldsymbol{\theta}_i^j - \hat{\boldsymbol{\theta}}_i^j)^T(-H(\hat{\boldsymbol{\theta}}_i^j))(\boldsymbol{\theta}_i^j - \hat{\boldsymbol{\theta}}_i^j)\big)d\boldsymbol{\theta}_i^j$, where $H(\hat{\boldsymbol{\theta}}_i^j)$ is the Hessian matrix of $\log P(\mathcal{D}_j|\boldsymbol{\theta}_i^j, M_i)$ calculated at $\hat{\boldsymbol{\theta}}_i^j$. When all the models are assumed to have the same probability *a priori*, the posterior model probabilities in (6.12), i.e. $P(M_i|\mathcal{D}), i = 1, ..., L$, are only dependent on the marginal probability of the observed data under each model in (6.13), i.e. $P(\mathcal{D}|M_i), i = 1, ..., L$.

6.3.3 Experiment Design by Bayesian Optimization

Bayesian Experiment Design (BED) has the potential to guide efficient search for desired materials by directing sequential search of "optimal" query points to approach the optimal solution [177]. Here, we employ the Expected Improvement (EI) [157] for single objective problems, and an extension of EI to guide the search to approach the Pareto front for multi-objective problems, namely the Expected Hyper-Volume Improvement (EHVI) [178]. Both EI and EHVI can balance exploration and exploitation up to some extent in guiding the search for optimal solutions.

A *major innovation* in our BED approach is that instead of assuming knowledge of the best predictive model in advance and updating this given predictive model based on the limited number of initial observed data and iterating the experiment design loop based on the updated model—an

approach that is taken in the literature—we consider the model uncertainty by including a class of potential predictive models for the task under study. By BMA, the experiment design step is performed based on the weighted average of these potential models. After performing the selected experiment, the new observed data from the experiment is used to update the (posterior) probability of all these potential predictive models. We can see that by taking this approach, as more experiments are done, the true predictive model is selected with a higher probability alongside accelerating the discovery of the material with the desired properties. We note that the proposed BMA also works in cases in which the feature sets are known or fixed but in which different model forms of the GPR—i.e. using different kernels—could potentially have different degrees of fidelity with regards to the available data.

For multi-objective problems, the EHVI under model averaging is

$$EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D}) = \int I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})P(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})d\boldsymbol{y} = \int I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})\sum_{i=1}^{L} P(M_i|\mathcal{D})P(\boldsymbol{y}|\boldsymbol{x},\mathcal{D},M_i)d\boldsymbol{y} = \sum_{i=1}^{L} P(M_i|\mathcal{D})EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D},M_i), \tag{6.17}$$

where $I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})$ denotes the hyper-volume improvement achieved by observing the outputs at $\boldsymbol{x}$, and $EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D},M_i)$ is the ordinary EHVI under model M_i. If the outputs are assumed to be independent $EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D},M_i)$ further simplifies to $\int I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})\prod_{j=1}^{q} P(y^j|\boldsymbol{x},\mathcal{D},M_i)d\boldsymbol{y}$. The optimal experiment to be performed next is $\boldsymbol{x}^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\ EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D})$, which is the one that maximizes the weighted average EHVI considering all the potential predictive models, again by the iteratively updated (posterior) model probabilities. The hyper-volume improvement $I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})$ is the increase in the hyper-volume of the dominated (objective) space achieved by adding the outputs at $\boldsymbol{x}$ to the observed data, i.e. $I_{\mathcal{H}}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D}) = \mathcal{H}(\boldsymbol{Y}\cup\boldsymbol{y}) - \mathcal{H}(\boldsymbol{Y})$. Without loss of generality, if we assume the goal is minimization of all the outputs, the hyper-volume dominated by a set of points $\boldsymbol{A}$ is defined as the volume of the dominated subspace by the points in A, i.e. $\mathcal{H}(\boldsymbol{A}) = \text{Volume}\left(\{\boldsymbol{s}\in\mathbb{R}^q|\boldsymbol{s}\prec\boldsymbol{r}\text{ and }\exists\,\boldsymbol{a}\in\boldsymbol{A}:\ \boldsymbol{a}\prec\boldsymbol{s}\}\right)$, where the domination rule is such that

$\boldsymbol{a} \prec \boldsymbol{b}$ if and only if $a^j \leq b^j$ for all $j = 1, ..., q$, and for at least one j, $a^j < b^j$. $\boldsymbol{r}$ is called a reference or anchor point and is a point dominated by all the possible output values (the whole output space).

For the special case of employing EI-based BED [157], the EI after observing data $\mathcal{D}$ can be computed under model averaging by:

$$\begin{aligned} EI(\boldsymbol{x}|\mathcal{D}) &= \int I(y|\boldsymbol{x}, \mathcal{D}) \sum_{i=1}^{L} P(M_i|\mathcal{D}) P(y|\boldsymbol{x}, \mathcal{D}, M_i) dy \\ &= \sum_{i=1}^{L} P(M_i|\mathcal{D}) \int I(y|\boldsymbol{x}, \mathcal{D}) P(y|\boldsymbol{x}, \mathcal{D}, M_i) dy = \sum_{i=1}^{L} P(M_i|\mathcal{D}) EI(\boldsymbol{x}|\mathcal{D}, M_i), \end{aligned} \tag{6.18}$$

where $I(y|\boldsymbol{x}, \mathcal{D})$ denotes the improvement achieved by observing the output of experiment $\boldsymbol{x}$, E represents expectation, and $EI(\boldsymbol{x}|\mathcal{D}, M_i)$ is the EI under model M_i. In this approach, the optimal experiment to be performed next is $\boldsymbol{x}^* = \underset{\boldsymbol{x} \in \chi}{\operatorname{argmax}}\, EI(\boldsymbol{x}|\mathcal{D})$. In the equations above, the improvement achieved by observing the output of experiment $\boldsymbol{x}$ is $I(y|\boldsymbol{x}, \mathcal{D}) = (y^* - y)_+$ when minimization is the goal, and $I(y|\boldsymbol{x}, \mathcal{D}) = (y - y^*)_+$ when maximization is the goal, where $(a)_+ = a$ if $a > 0$ and is zero otherwise, and y^* denotes the best (lowest/highest for minimization/maximization problems) output observed so far, i.e. the best output in $\mathcal{D}$.

The algorithm for our proposed Bayesian Optimization under Model Uncertainty (BOMU) framework is shown in Algorithm 3 and the overall framework for autonomous materials discovery is shown in Figure 6.2. In Algorithm 3, for the single-objective case, $u(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}}, M_i)$ and $u(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}})$ correspond to $EI(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}}, M_i)$ and $EI(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}})$, and for the multi-objective case correspond to $EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}})$ and $EI_{\mathcal{H}}(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}}, M_i)$, respectively.

Algorithm 3 Bayesian Optimization under Model Uncertainty

1: Initialize $\mathcal{D}_0$

2: **for** `n=0,1,...` **do**

3: Update statistical model(s), M_i

4: Compute acquisition function u with model averaging:

$$u\left(\mathbf{x}|\mathcal{D}_{\mathrm{n}}\right) = \sum_{i=1}^{L} P(M_i|\mathcal{D}_{\mathrm{n}}) u(\boldsymbol{x}|\mathcal{D}_{\mathrm{n}}, M_i)$$

5: Select new $\mathbf{x}_{\mathrm{n}+1}$ by optimizing acquisition function u:

$$\mathbf{x}_{\mathrm{n}+1} = \arg\max_{\mathbf{x}\in\chi} u\left(\mathbf{x}|\mathcal{D}_{\mathrm{n}}\right)$$

6: Query blackbox function f to obtain y_{n+1}

7: Augment data $\mathcal{D}_{\mathrm{n}+1} = \{\mathcal{D}_{\mathrm{n}}, (\mathbf{x}_{\mathrm{n}+1}, y_{\mathrm{n}+1})\}$

8: **if** stopping criteria reached **then**

9: **break**

10: **end if**

11: **end for**

6.3.4 Results and Discussion

Because of their rich chemistry and the wide range of values of their properties [179], MAX phases constitute an adequate material system to test simulation-driven, specifically DFT calculations, materials discovery frameworks. [180] used the MAX phases with M_2AX stoichiometry to deploy and test different Bayesian Optimization schemes. In this work, we use the same system to test the proposed framework.

The MDS for this work is composed of conventional MAX phases with M_2AX and M_3AX_2

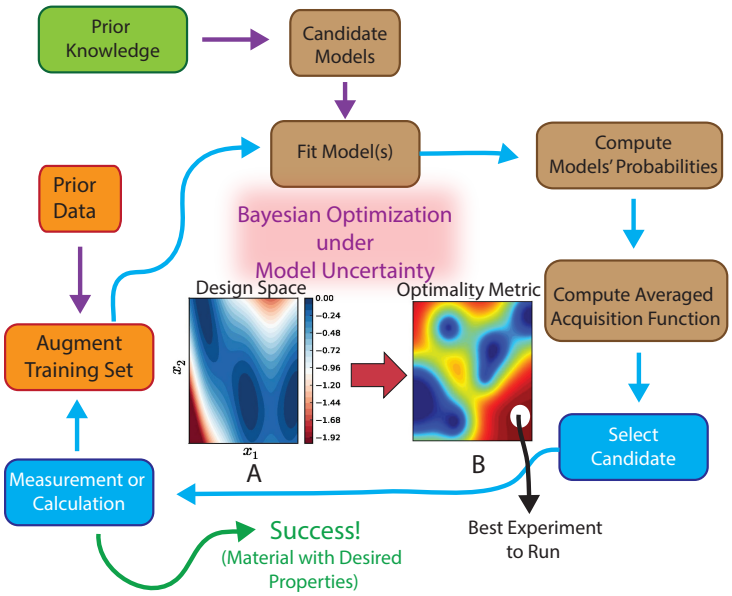


Figure 6.2: Schematic of the proposed framework for an autonomous, efficient materials discovery system as a realization of Bayesian Optimization under Model Uncertainty (BOMU).

stoichiometries. Here M $\in \{Sc, Ti, V, Cr, Zr, Nb, Mo, Hf, Ti\}$; A $\in \{Al, Si, P, S, Ga, Ge, As, Cd, In, Sn, Tl, Pd\}$; and X $\in \{C, N\}$. This results in 216 M_2AX and 216 M_3AX_2 phases. Since we are testing a materials discovery framework, we found it convenient to determine the ground truth of the system beforehand and the mechanical properties of these systems were thus determined before deploying the BOMU framework —our framework has been incorporated into a high-throughput workflow automation tool using the scikit-learn [181] toolbox.

The problem was formulated with the goal of identifying the material/materials with i) the maximum bulk modulus K; ii) the minimum shear modulus G; and iii) the maximum bulk modulus and minimum shear modulus. The cases of i) the maximum bulk modulus K; ii) the minimum shear modulus G are designed as single-objective optimization problems. The third problem which seeks to identify the materials with the maximum bulk modulus and minimum shear modulus (iii) is designed as a multi-objective problem.

The complete experimental details, results, and discussion can be found in [166]. Here, we only include selected results and discussions from the published paper.

In this work, prior knowledge is available before starting the materials discovery task. The prior knowledge is in terms of feature sets that are likely to have effects on the materials properties of interest. Six feature sets are constructed based on domain knowledge and the physical or chemical properties they represent.

For each of the targets (maximizing K, minimizing G, as well as maximizing-K/minimizing G) we carried out the sequential experiment design by maximizing the EI or EHVI based on predictive models using single feature sets or BMA using all the feature sets accounting for their probability through first-order (BMA_1) and second-order (BMA_2) Laplace approximation. The budget for the optimal design was set at $\approx 20\%$ of the MDS, i.e 80 materials or calculations.

Figure 6.3 shows the comparison of the average performance of both the first-order and second-order BMA over all initial data set instances with the best performing model (F_2) and worst performing model (F_6). Note that the best and worst performing models are not known *a priori* in practice. In the Figure, for the test problem to find the MAX phase with the maximum bulk or minimum shear modulus, the maximum or minimum values found in the experiment design iterations averaged over all initial data set instances starting with 20 initial points are shown. The dotted line in the figure indicates the maximum bulk modulus = 300 GPa or minimum shear modulus = 10.38 GPa that can be found in the MDS. It can be seen that both the first-order and second-order BMA performance in identifying the maximum bulk or minimum shear modulus is consistently close to the best model (F_2).

In Figs. 6.4(a) and 6.4(b), the average model coefficients (posterior model probabilities) of the GPR models based on different feature sets over all instances of initial data set are shown with the increasing number of calculations for BMA_1 and BMA_2, respectively. It can be seen that these model coefficients from BMA may guide automatic selection of the best model and feature set F_2.

Note that without having actually gone through the experiment design loop, one could not know *a priori*, that using F_6 will result in not arriving at the desired material within a reasonable budget with a very high probability. The results here and in [166] show that if one were to just select a feature set even using domain knowledge, one may or may not select a good model. However, if

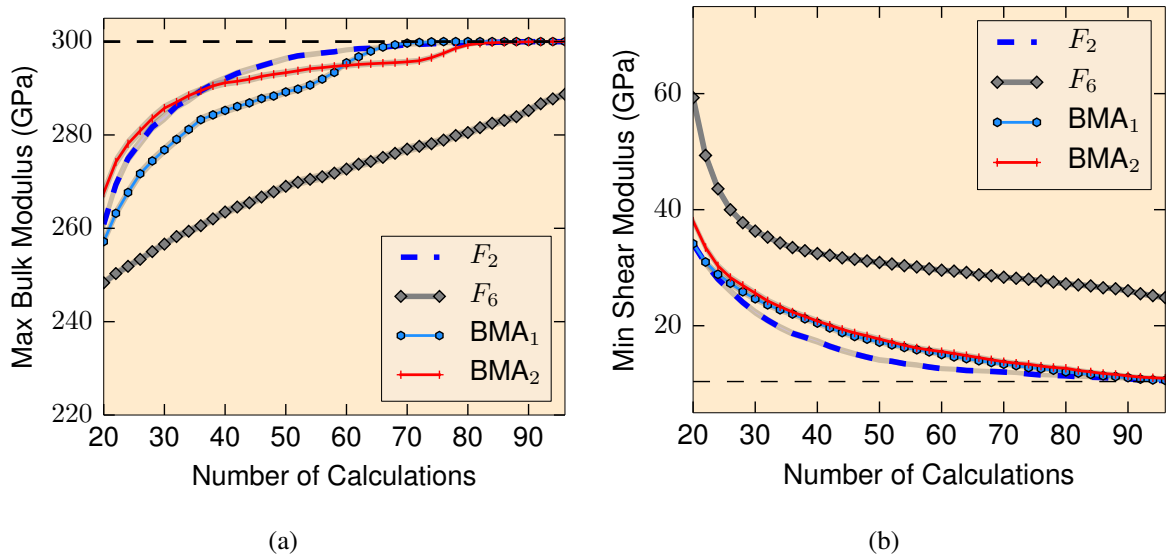
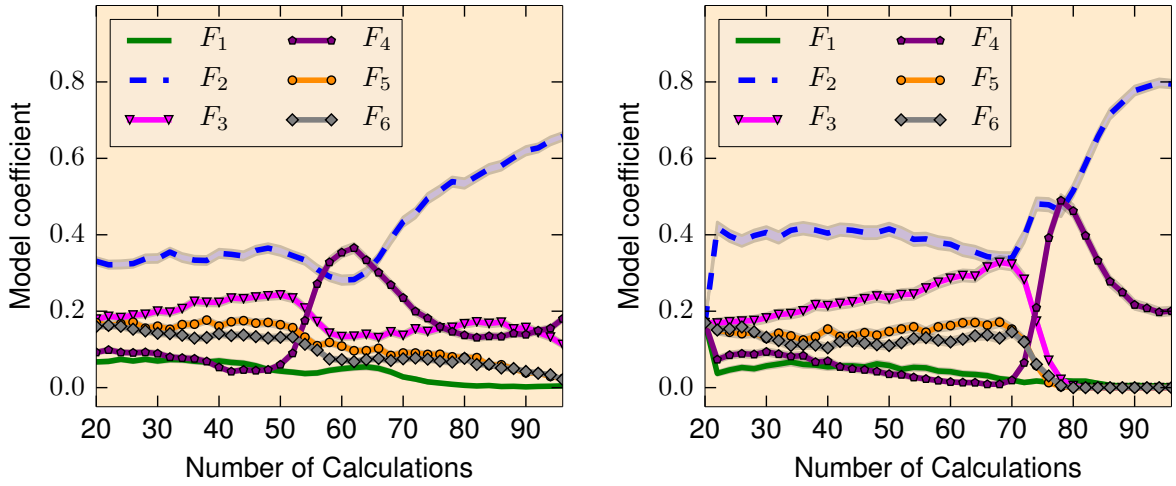


Figure 6.3: Representative results for single objective optimization starting with 20 initial points using the best model F_2, worst model F_6, BMA$_1$ and BMA$_2$: a) Average maximum bulk modulus discovered, b) Average minimum shear modulus discovered

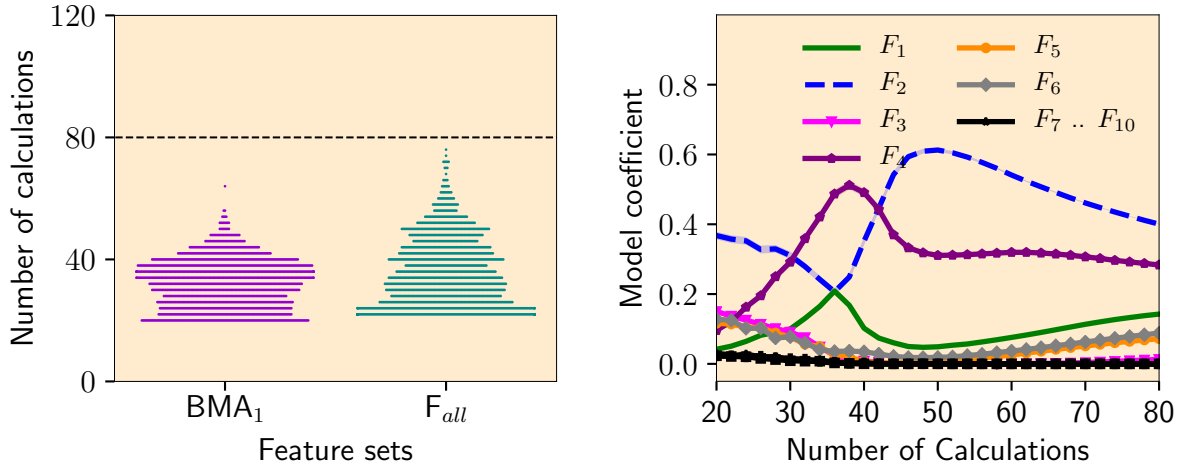
one were to use the BMA approach, either BMA$_1$ or BMA$_2$, the probability of successfully arriving at the material with desired properties, is very high, since the BMA approach auto-selects the best model (more corresponding results available in [166]).

To further showcase the utility of our proposed approach, we simulate a high-dimensional case by adding 16 non-informative random features, which we compose into subsets F_7, F_8, F_9, and F_{10} of four features each. We carry out two types of calculation using the larger set of 29 (13+16) features. First, we use the BMA$_1$ approach to find material with maximum K using models based on F_1,...F_{10}; and we use the regular EGO-GP framework to find the material with maximum K using all 29 features (F$_{all}$). The results are plotted in Figure 6.5. We see in Figure 6.5a, that in this case (an actual high dimensional case with a number of non-informative random features), the BMA approach outperforms using all features together. Additionally, tracking the model probabilities as in Figure 6.5b, shows us that the BMA approach effectively picks up F_2 set as the best model, rejects the random feature sets F_7, ...F_{10} (average model probabilities are negligible) and performs better than using F$_2$ standalone (corresponding result available in [166]).



(a) BMA_1 (b) BMA_2

Figure 6.4: Average model probabilities for maximizing bulk modulus using a) BMA_1 and b) BMA_2



(a) (b)

Figure 6.5: Representative results for single objective optimization – minimization of shear modulus for the case of 29 features: a) swarm plots indicating the distribution of the number of calculations required for convergence to the optimal solution using BMA_1 and F_{all} b) average model probabilities for maximizing bulk modulus using BMA_1 and F_{all}.

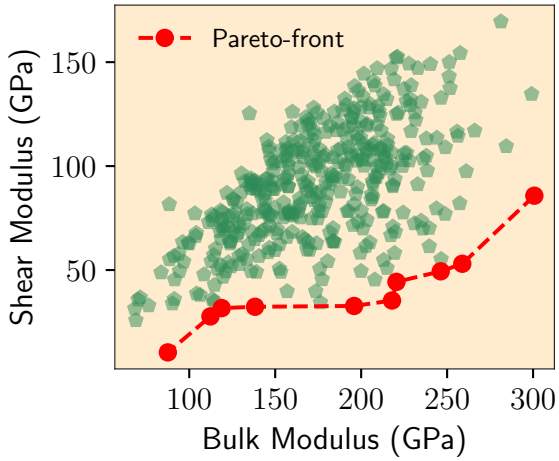


Figure 6.6: The Pareto optimal points in the materials property space are marked in red corresponding to the criterion of maximizing bulk modulus and minimizing shear modulus simultaneously. The Pareto set for the MDS consists of 10 points as indicated in red.

We now consider multi-objective experiment design to optimize two objectives at the same time: maximizing bulk modulus and minimizing shear modulus. One should note that in our analysis we have already calculated the responses of bulk and shear modulus as materials properties for all the feasible points in the MDS to have the ground truth to compare different models for experiment design. Generally in practice, no knowledge of the responses exists unless one performs all the possible experiments exhaustively. Consequently, none of this information is used in our experiment design procedures. Figure 6.6 illustrates all the data points in the objective space of materials properties (in green). It can be seen that in this case there does not exist a single optimal solution, and in fact there are ten *Pareto* optimal points comprising the *Pareto front* which is highlighted in red in the figure. Specifically, the Pareto front here is the 1-dimensional design curve over which any improvement in one material property (i.e bulk modulus K) is only achieved through a corresponding sacrifice of another property (here, shear modulus G).

Figure 6.7 depicts the average performance of the best (F_2) and worst (F_1) models as well as the first- and second-order BMA in finding the true Pareto optimal points versus the number of calculations, starting from 10 initial points. Similar to single-objective problems, multi-objective

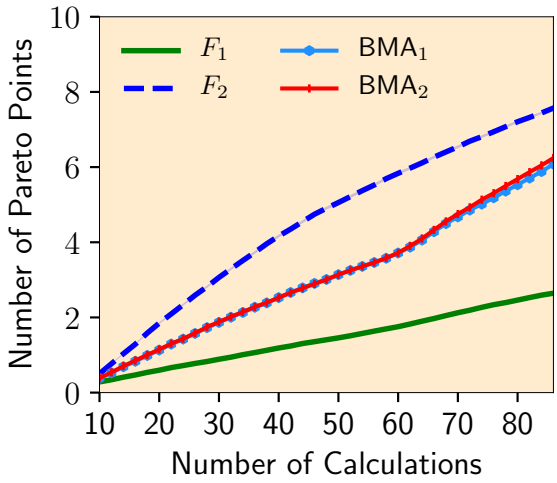


Figure 6.7: Average number of true Pareto optimal points found over all initial data set instances for single models, BMA$_1$, and BMA$_2$.

experiment design based on F_2 consistently has the best performance; i.e. it identifies more true Pareto optimal points faster (with smaller budget). Both BMA approaches' performances are consistently in the range of the first best single model's performances.

From the results in this section and in [166], we can see that for single-objective experiment design, the performance of the first-order BMA is sometimes slightly better than the second-order BMA. On the other hand, the model probabilities in the second-order BMA are more robust, and at any calculation number (sequential experiment iteration), the average posterior probability over all the initial data set instances of the best model in terms of experiment design performance is higher than the other models. The reason is that second-order Laplace approximation, unlike the first-order one, does not rely solely on the fitted values of the parameters of the GPR model to calculate the model probability. In fact, it approximates the model probability by integrating a local expansion of the marginal likelihood over a neighborhood of the fitted parameters values, which may dampen the fluctuations of the fitted values between different sequential experiment iterations. For the multi-objective case, the second-order BMA is slightly better than first-order BMA in terms of both experiment design performance and robustness of identifying the best model in terms of experiment design performance.

A final remark on the feature sets is that in our analysis, they are chosen *a priori* based on domain knowledge. We do not claim that the considered feature sets are among the best possible feature sets for our experiment design problems. We are rather using these to showcase the applicability of the BOMU framework in real-world experiment design problems, where the best model or feature set is often not known, and only a set of possible models might exist based on domain knowledge. The power of BOMU is that it incorporates the uncertainty over the possible model space, instead of relying on a single model that is selected based on limited initial available data.

7. BAYESIAN PROPER ORTHOGONAL DECOMPOSITION FOR LEARNABLE REDUCED-ORDER MODELS WITH UNCERTAINTY QUANTIFICATION

7.1 Introduction

Designing and/or controlling complex systems in science and engineering relies on appropriate mathematical modeling of systems dynamics. Classical differential equation based solutions in applied and computational mathematics are often computationally demanding. Recently, the connection between reduced-order models of high-dimensional differential equation systems and surrogate machine learning models has been explored. However, the focus of both existing reduced-order and machine learning models for complex systems has been how to best approximate the high fidelity model of choice. Due to high complexity and often limited training data to derive reduced-order or machine learning surrogate models, it is critical for derived reduced-order models to have reliable uncertainty quantification at the same time. In this Chapter, we propose such a novel framework of Bayesian reduced-order models naturally equipped with uncertainty quantification as it learns the distributions of the parameters of the reduced-order models instead of their point estimates. The developed method has the capability of embedding physics constraints when learning the surrogate reduced-order models, a desirable feature when studying complex systems in science and engineering applications where the available training data are limited.

Machine learning and artificial intelligence (ML/AI) have been revolutionizing modeling and decision-making in many real-world applications [182]. If generalizable predictive models can be learned, typically from "big" data, ML/AI can greatly help effective and efficient decision making. However, when facing complex natural and engineered systems, where available data of observations are small with respect to the system complexity, deriving generalizable ML models can be challenging. On the other hand, in applied and computational mathematics, research in simulating high-dimensional complex systems has been studied extensively with rich knowledge in fundamental physics principles, such as conservation laws and other governing equations. Nonetheless,

it is often computationally expensive to simulate high-dimensional systems dynamics, typically by solving the corresponding Ordinary or Partial Differential Equation Systems (ODE/PDEs). Many recent research efforts have been made to develop ML methods to speed up computational simulations based on differential equation systems.

For example, neural networks have been used as (black-box) surrogates for physical systems [183, 184], and have recently gained renewed interest [185, 186] due to widespread availability of more powerful computational resources. Physics-informed neural networks (PINN) [185] represent one of such models where the input to the neural network is the spatial coordinates (and also time if time-dependent) and the output is the predicted output field(s). In PINNs, the physics principles are added via regularization terms in addition to the reconstruction loss for training the surrogate to encourage it to respect the underlying governing equations and the initial/boundary conditions with the help of automatic differentiation. PINNs have been recently extended [187] by employing Bayesian neural networks, i.e. placing a prior on the network weights and calculating an approximate posterior, to have a notion of uncertainty estimate. The Bayesian version of PINNs can only use samples from the boundary conditions and not full knowledge of it. Also, the experiments in [187] have shown that the training of Bayesian PINNs can be challenging where simpler variational approximations do not usually work and they require the more computationally complex Hamiltonian Monte Carlo approximation in order to result in satisfactory performances. In [186], Bayesian convolutional neural networks for image to image regression are used as a surrogate model for flow through porous media. The approach taken there lacks any specific mechanism to enforce boundary conditions. All these methods lack an interpretable lower-dimensional embedding, need retraining if boundary/initial conditions are changed, and still require a quite significant amount of data for training. Other works like [188, 189] assume that all the underlying governing equations are fully known and utilize them to train a neural network to imitate them.

In this Chapter, motivated by recent efforts to derive reduced-order models of high-fidelity differential equation systems by physics-based ML to embed physics constraints [190], we leverage Bayesian learning to develop a new framework of Bayesian reduced-order models (ROMs). Be-

sides searching for reduced-order models that best approximate the high-fidelity differential equation solutions, Bayesian ROMs emphasize naturally-equipped uncertainty quantification capability, which is critical when designing and controlling complex systems in science and engineering often with little-to-no observed data, to enable reliable estimates of prediction confidence for robust decision making. Moreover, when learning reduced-order models of differential equation systems, the underlying scientific principles can be naturally incorporated as shown in [190].

There exist a wide variety of model reduction methods [191, 192, 193] that search for the best low-dimensional approximations of an underlying high-fidelity model, which is typically a high-dimensional system of ordinary differential equations or a system of equations stemming from the discretization of partial differential equations characterizing the corresponding systems dynamics. In this Chapter, we focus on reduced-order models based on the proper orthogonal decomposition (POD) [194] as they are closely related to subspace learning in ML/AI. In addition, the projection-based POD can be derived with embedded physics constraints, including system geometry, system configuration, initial conditions, and boundary conditions [190]. In particular, we develop *learnable* Bayesian POD (**BayPOD**). In BayPOD, we propose to simultaneously learn the distributions of both the POD projection bases and the mapping from the system input parameters to the projected scores/coefficients from "snapshots," solutions computed with the high-fidelity model for different inputs, which can include both the settings for the parameters of the full (high-fidelity) model and initial or boundary conditions.

Figure 7.1 provides a schematic illustration of BayPOD, which leverages the subspace learning and regression models into one unified Bayesian learning framework to help reliably predict high-dimensional systems dynamics/fields as quantifies of interest with significantly improved scalability and computational efficiency compared to the original high-dimensional ODE/PDE solvers. More critically, the learned BayPOD models, due to its generative nature, can provide reliable uncertainty estimates of predicted systems dynamics in different setups, which will be the enabler of optimal and adaptive decision making when studying and intervening complex systems of interest.

Compared to the existing reduced-order models, our BayPOD has the following advantages:

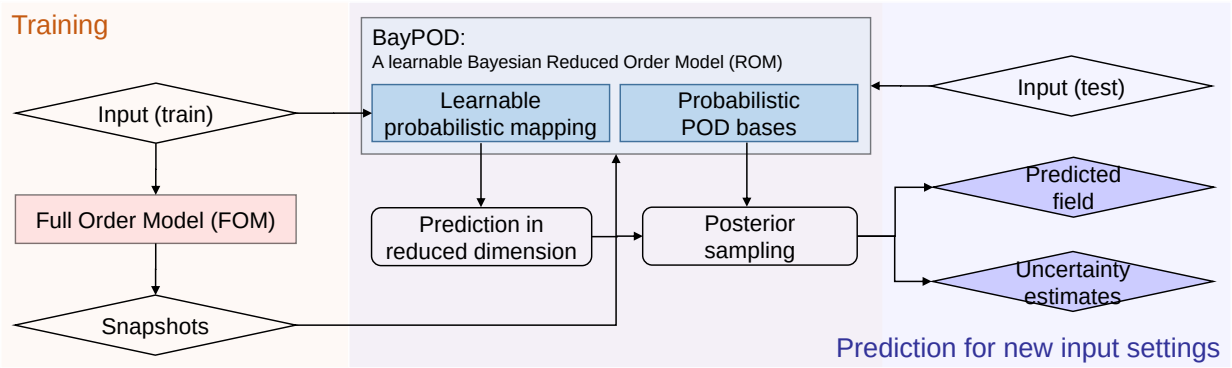


Figure 7.1: Schematic diagram of BayPOD at training and for prediction. Inputs can include settings for the parameters of the full (high-fidelity) model and initial or boundary conditions.

- Our framework provides a unified way for learning POD basis and coefficients without resorting to multiple independent steps, as originally implemented in [190].
- We can quantify the uncertainty about field prediction for new inputs through posterior distributions.
- By incorporating prior distributions, the POD basis parameters are regularized to mitigate the impacts of high-dimensional snapshots with small sample size.
- Flexible models, such as neural networks (NNs), can be integrated for mapping from systems inputs to POD coefficients when needed, using amortized variational inference.
- Our BayPOD enables Bayesian experimental design with reduced-order models based on scientific principles, instead of "black-box" surrogate models.

The organization of the rest of the Chapter is as follows. We first briefly review the background of POD and its machine learning extensions with physics constraints. We then present BayPOD and the corresponding inference algorithms. In Sections 7.3.1 and 7.3.2, case studies of predicting the temperature field of a heated rod and the pressure field around an airfoil are performed with both prediction and uncertainty quantification performance evaluation.

7.2 Methods

7.2.1 Proper Orthogonal Decomposition (POD)

Consider a system that maps an input onto a physical field such as pressure, temperature, stress, strain, etc. The physical field is the quantity of interest that we aim to predict. Denote a field as a function $f : \mathcal{X} \times \mathcal{T} \times \mathcal{P} \to \mathbb{R}$, with the spatial domain $\mathcal{X}$, time domain $\mathcal{T}$, and input domain $\mathcal{P}$. The field f varies in space and time, and depends on the input of the system. Given the observed data $\mathcal{D} \subset \{f(\boldsymbol{x}, t; \boldsymbol{p}) | \boldsymbol{x} \in \mathcal{X}, t \in \mathcal{T}, \boldsymbol{p} \in \mathcal{P}\}$, we focus on learning approximate models f that respect the underlying physical constraints of the system.

Proper orthogonal decomposition (POD) is one of the most widely used model reduction methods which computes an expansion basis that enables a low-dimensional representation of the high-dimensional system state [190]. Consider the field $f(\cdot, t; \boldsymbol{p})$ at time $t \in \mathcal{T}$ and with input $\boldsymbol{p} \in \mathcal{P}$. To calculate the POD basis, we introduce the finite-dimensional approximation $\boldsymbol{f}(t; \boldsymbol{p}) \in \mathbb{R}^{n_x}$ of $f(\cdot, t; \boldsymbol{p})$, where n_x is the dimension of the finite-dimensional discretization of the spatial domain. The approximate field $\boldsymbol{f}(t; \boldsymbol{p})$ is referred to as a *snapshot*, and it can be sensed data or a computational solution generated by a numerical model. The POD basis is computed using many such collected snapshots.

Let $\{\boldsymbol{f}(t_i; \boldsymbol{p}_j) | i = 1, ..., n_t, j = 1, ..., n_p\}$ be the set of $n_s = n_t n_p$ snapshots at n_t different time instances $\{t_1, ..., t_{n_t}\} \subset \mathcal{T}$ and for n_p different inputs $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_{n_p}\} \subset \mathcal{P}$. The POD bases are then obtained by singular value decomposition (SVD) of the snapshot matrix $F = \left[\boldsymbol{f}(t_i; \boldsymbol{p}_j)\right]_{i,j} \in \mathbb{R}^{n_x \times n_s}$, which contains the snapshot vectors as its columns. More precisely, the SVD can be written as

$$F = V \Sigma W,$$

where the columns of the matrices $V \in \mathbb{R}^{n_x \times n_s}$ and $W \in \mathbb{R}^{n_s \times n_s}$ are the left and right singular vectors of F, respectively. The POD basis of dimension K , $V_K = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]$, is then defined as the K left singular vectors of F that correspond to the k largest singular values, where $K << n_x$.

7.2.2 Physical Fields in the POD Basis

After learning the POD basis from snapshot data, any field f can be approximated by a linear expansion as:

$$\tilde{\boldsymbol{f}}(t;\boldsymbol{p}) = \sum_{k=1}^{K} \boldsymbol{v}_k \alpha_k(t;\boldsymbol{p}), \tag{7.1}$$

where $\alpha_k(t;\boldsymbol{p})$ is the POD expansion coefficients and $\tilde{\boldsymbol{f}}(t;\boldsymbol{p})$ is the approximation of the field $f(\cdot,t;\boldsymbol{p})$ at time t and input $\boldsymbol{p}$. The POD expansion coefficients can be calculated as $\alpha_k(t;\boldsymbol{p}) = \boldsymbol{v}_k^T \boldsymbol{f}(t;\boldsymbol{p})$, for $k \in \{1, ..., K\}$.

The linear representation (7.1) provides a mechanism for embedding physical constraints. An approach to embed physical constraints into POD representation is by considering an alternative representation to (7.1) as:

$$\tilde{\boldsymbol{f}}(t;\boldsymbol{p}) = \bar{\boldsymbol{f}} + \sum_{k=1}^{K} \bar{\boldsymbol{v}}_k \alpha_k(t;\boldsymbol{p}), \tag{7.2}$$

where $\bar{\boldsymbol{f}}$ is a *particular solution*. As an example, the particular solution $\bar{\boldsymbol{f}}$ is chosen to satisfy a particular set of prescribed inhomogeneous boundary conditions and the POD bases $\bar{\boldsymbol{v}}$ are defined so that they satisfy homogeneous boundary conditions.

7.2.3 Learning POD Coefficients

Recently, machine learning methods have been employed to learn a surrogate model for the map $\boldsymbol{\alpha} : \mathcal{P} \to \mathcal{A}$ from inputs $\boldsymbol{p} \in \mathcal{P}$ to the POD coefficients $\boldsymbol{\alpha}(\boldsymbol{p}) \in \mathcal{A}$, where $\boldsymbol{\alpha}(\boldsymbol{p}) = [\alpha_1(\boldsymbol{p}), ..., \alpha_K(\boldsymbol{p})]$ and we assume inputs $\boldsymbol{p} = [p_1, ..., p_m]$ are m-dimensional system parameters [190]. In the first step, we collect the inputs corresponding to the snapshots in a matrix $P \in \mathbb{R}^{n_s \times m}$, and their corresponding POD coefficients in a matrix $A \in \mathbb{R}^{n_s \times K}$. Then, input and output data are divided into training and test sets, and the map $\boldsymbol{\alpha} : \mathcal{P} \to \mathcal{A}$ is learned from the training data by applying supervised machine learning methods such as neural networks, decision trees or k-nearest neighbors regression model [190].

7.2.4 Bayesian POD

In this section, we introduce our framework of Bayesian reduced-order models, BayPOD, which simultaneously learns the distributions of both POD projection bases and mapping from system inputs to projection coefficients. BayPOD is a Bayesian matrix factorization framework for simultaneously learning POD bases together with the relationship between input parameters and POD coefficients. The modeling of mapping from inputs to coefficients can be flexible. In this Chapter, we focus on linear parameter models (BayPOD-LM) first and then extend it to neural network models (BayPOD-NN) with amortized variational inference.

7.2.5 BayPOD – A Generative POD Model

We start by modeling the homogeneous field $\tilde{\boldsymbol{f}}$ in (7.1) using a multivariate normal distribution. The framework can be readily extended to (7.2) by adding the particular solution $\bar{\boldsymbol{f}}$.

Let $\tilde{f}_{sx}$ denote the field response for snapshot $s \in \{1, 2, ..., n_s\}$ at the spatial point $x \in \{1, 2, ..., n_x\}$. We model this response as a normally-distributed random variable:

$$\tilde{f}_{sx} \sim \mathrm{N}(\boldsymbol{u}_x^T \boldsymbol{\alpha}_s, \gamma_x^{-1}), \tag{7.3}$$

where $\boldsymbol{u}_x = [u_{x1}, ..., u_{xK}] \in \mathbb{R}^K$ is the K-dimensional POD basis vector at position x and $\boldsymbol{\alpha}_s = [\alpha_{s1}, ..., \alpha_{sK}] \in \mathbb{R}^K$ represents the K POD coefficients for snapshot s. The variance γ_x^{-1} can be considered as the model uncertainty at position x.

We place independent zero-mean normal priors on POD basis and coefficients:

$$\begin{aligned} \boldsymbol{u}_x &\sim \mathrm{N}(0, I_K), \\ \boldsymbol{\alpha}_s &\sim \mathrm{N}(0, \gamma_\alpha^{-1} I_K), \end{aligned} \tag{7.4}$$

where I_K is the identity matrix, and γ_α is the precision parameter for $\boldsymbol{\alpha}_s$. Note that k indexes the dimension of subspace (POD bases/factors/PCs). Employing the priors in (7.4) has multiple benefits. First, by placing zero-mean priors on $\boldsymbol{u}$ and $\boldsymbol{\alpha}$, we ensure that the marginal distribution

of $\tilde{\boldsymbol{f}}$ is zero mean, and thus physical constraints can be applied through the particular solution. Second, normal priors enhance the robustness of our model in the presence of small sample size data, as they play a role similar to ridge regularization. Finally, by using identity covariance matrix for POD basis $\boldsymbol{u}$ in the prior distribution, we aim to reduce the unidentifiability of $\boldsymbol{u}$ from $\boldsymbol{\alpha}$ in the model. To complete the model, we place conjugate gamma distributions over the position and coefficient precision parameters:

$$\gamma_\alpha, \gamma_x \sim \text{Gamma}(1, 1). \tag{7.5}$$

7.2.5.1 Inference model

A primary goal of model reduction is to predict the system response to new input parameters by leveraging the learned basis vectors. We attain this goal by introducing an inference (recognition) network, widely used in variational inference literature [195, 196, 118, 197].

In variational inference framework, we introduce variational distributions $q(\cdot)$ over model parameters as approximations for intractable posterior distributions. For our Bayesian reduction model, to simplify deriving the variational parameters, we assume the following independence structure for variational distributions:

$$q(\boldsymbol{u}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = q(\boldsymbol{u})q(\boldsymbol{\alpha})q(\boldsymbol{\gamma}). \tag{7.6}$$

To establish amortized inference of POD coefficients $\boldsymbol{\alpha}_s$ for $s \in \{1, ..., n_s\}$, we define their variational distributions as

$$q(\boldsymbol{\alpha}_s) = \text{N}(\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}), \Sigma_{\boldsymbol{w}}(\boldsymbol{p})), \tag{7.7}$$

where $\boldsymbol{\mu}_{\boldsymbol{w}}$ and $\Sigma_{\boldsymbol{w}}$ are mean and covariance matrix which take the form of some mapping with weights $\boldsymbol{w}$ from input parameters $\boldsymbol{p}$. Hence, for new input parameters $\boldsymbol{p}^*$, the variational posterior mean $\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}^*)$ can be considered as an estimate of the POD coefficients.

Finally, to exploit the conjugate priors, we let the variational posteriors for POD basis and

precision parameters to be normal and gamma distributions, respectively:

$$
\begin{aligned}
q(\boldsymbol{u}_x) &= \mathrm{N}(\boldsymbol{\mu}_x, \Sigma_x), \\
q(\gamma_x) &= \text{Gamma}(\lambda_x, 1/r_x), \\
q(\gamma_\alpha) &= \text{Gamma}(\lambda_\alpha, 1/r_\alpha).
\end{aligned} \tag{7.8}
$$

To obtain the optimal variational parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \Sigma, \boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{r}, \boldsymbol{w}\}$, we minimize the KL-divergence between the variational posteriors and the true posteriors, or equivalently maximize the evidence lower bound (ELBO) of the marginal log-likelihood [117, 198]:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\Theta}) &= \mathbb{E}_{q(\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\gamma})}\Big[\log \frac{p(\tilde{\boldsymbol{f}}|\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\gamma})p(\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\gamma})}{q(\boldsymbol{u},\boldsymbol{\alpha},\boldsymbol{\gamma})}\Big], \\
&\leq \log p(\tilde{\boldsymbol{f}}).
\end{aligned} \tag{7.9}
$$

Below, we present the update equations for the variational parameters.

Update $\boldsymbol{u}$: Using the conjugacy property of normal distributions, we can derive the closed form of variational parameters for $\boldsymbol{u}_x$ as follows:

$$
\begin{aligned}
\Sigma_x &= \Big(< \gamma_x > \sum_{s=1}^{n_s} < \boldsymbol{\alpha}_s \boldsymbol{\alpha}_s^T > + I_K \Big)^{-1}, \\
\boldsymbol{\mu}_x &= \Sigma_x \big[< \gamma_x > \sum_{s=1}^{n_s} \tilde{f}_{sx} < \boldsymbol{\alpha}_s > \big], \\
< \gamma_x > &= \lambda_x / r_x, \\
< \boldsymbol{\alpha}_s > &= \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}), \\
< \boldsymbol{\alpha}_s \boldsymbol{\alpha}_s^T > &= \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}_s)\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}_s)^T + \Sigma_{\boldsymbol{w}}(\boldsymbol{p}_s),
\end{aligned} \tag{7.10}
$$

where $< \cdot >$ denotes expectation with respect to the variational distributions.

Update $\boldsymbol{\gamma}$: Similar to $\boldsymbol{u}$, we exploit the conjugacy to obtain the variational parameters for both

γ_x and γ_α. For γ_x, we have:

$$\begin{aligned} \lambda_x &= 1 + n_s/2, \\ r_x &= 1 + \frac{1}{2}\sum_{s=1}^{n_s} < (\tilde{f}_{sx} - \boldsymbol{u}_x^T \boldsymbol{\alpha}_s)^2 >, \end{aligned} \tag{7.11}$$

where the expectations in the second line can be calculated using the following equations:

$$\begin{aligned} < \boldsymbol{u}_x > &= \boldsymbol{\mu}_x, \\ < \boldsymbol{u}_x \boldsymbol{u}_x^T > &= \boldsymbol{\mu}_x \boldsymbol{\mu}_x^T + \Sigma_x \\ < \boldsymbol{\alpha}_s^T A \boldsymbol{\alpha}_s > &= \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}_s)^T A \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}_s) + tr(A\Sigma_{\boldsymbol{w}}(\boldsymbol{p}_s)). \end{aligned} \tag{7.12}$$

Similarly, for γ_α, we can update the variational distribution's parameters as:

$$\begin{aligned} \lambda_\alpha &= 1 + \frac{n_s K}{2}, \\ r_\alpha &= 1 + \frac{1}{2}\sum_{s=1}^{n_s} < \boldsymbol{\alpha}_s^T \boldsymbol{\alpha}_s > . \end{aligned} \tag{7.13}$$

Variational inference alternates the updates of these model parameters and parameters of the mapping by: $\log q(\theta_i) \propto \mathbb{E}_{-i}[\log p(\mathcal{D}, \theta)]$. The main implementation difference with different mappings is to update $\boldsymbol{\alpha}$ according to the model.

Update $\boldsymbol{\alpha}$: To update the parameters of the mapping from the inputs to the variational distribution, we optimize the evidence lower-bound with respect to the parameters of $q(\boldsymbol{\alpha})$, where the objective function can be expressed as:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\boldsymbol{\alpha})q(\boldsymbol{u})q(\boldsymbol{\gamma})}\Big[\log \prod_{s,x} N(\tilde{f}_{sx}; \boldsymbol{u}_x^T \boldsymbol{\alpha}_s, \gamma_x^{-1})\Big] \\ &- \mathbb{E}_{q(\boldsymbol{u})q(\boldsymbol{\gamma})}\Big[\mathrm{KL}\big[N(\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{p}), \Sigma_{\boldsymbol{w}}(\boldsymbol{p}))||N(0, \gamma_\alpha^{-1} I_K)\big]\Big]. \end{aligned} \tag{7.14}$$

Note here that the functional relationships from the input parameters to the variational distribution

is through $\boldsymbol{\mu_w}(\boldsymbol{p})$ and $\Sigma_{\boldsymbol{w}}(\boldsymbol{p})$, which can be modeled flexibly with different complexity levels to balance the model expressiveness and computational as well as sample complexity. In this Chapter, we illustrate two implementation options with simple linear models and non-parametric neural networks.

7.2.5.2 Bayesian POD with linear mappings (BayPOD-LM)

In the first scenario, we employ a linear mapping from the input $\boldsymbol{p}$ to $\boldsymbol{\alpha}$, which we call **BayPOD-LM**, with

$$q(\boldsymbol{\alpha}_s|\boldsymbol{p}_s) = N(\boldsymbol{\mu_w}(\boldsymbol{p}_s), \Sigma_{\boldsymbol{w}}(\boldsymbol{p}_s)) = N(\boldsymbol{W}\boldsymbol{p}_s, \Sigma_{\boldsymbol{\alpha}}), \tag{7.15}$$

where $\boldsymbol{W} \in \mathbb{R}^{K\times m}$, and m is the system parameters' cardinality. Note that here, the mean in (7.7) is a linear function of the input parameters and the covariance is shared across different inputs. For BayPOD-LM, we can express (7.14) by keeping only the terms depending on $\boldsymbol{W}$ and $\Sigma_{\boldsymbol{\alpha}}$ as:

$$\begin{aligned}
\mathcal{L} &= \sum_{s=1}^{n_s} < \gamma_\alpha > \boldsymbol{p}_s^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{p}_s + \frac{n_s}{2} \log |\Sigma_{\boldsymbol{\alpha}}| \\
&- \sum_{s,x} \frac{< \gamma_x >}{2} \big[\boldsymbol{p}_s^T \boldsymbol{W}^T < \boldsymbol{u}_x \boldsymbol{u}_x^T > \boldsymbol{W} \boldsymbol{p}_s \\
&+ tr(< \boldsymbol{u}_x \boldsymbol{u}_x^T > \Sigma_{\boldsymbol{\alpha}}) - 2\tilde{f}_{sx} \boldsymbol{p}_s^T \boldsymbol{W}^T < \boldsymbol{u}_x > \big],
\end{aligned} \tag{7.16}$$

where $|\cdot|$ is the determinant. We can calculate the gradients of the objective function in (7.16) with respect to $\boldsymbol{W}$ and $\Sigma_{\boldsymbol{\alpha}}$ in closed form. Hence, we have the following update equations integrated into the inference procedure of general BayPOD:

Update $\boldsymbol{\alpha}$: To update the parameters of the linear model which outputs $\boldsymbol{\alpha}$:

$$\begin{aligned}
\Sigma_{\boldsymbol{\alpha}} &= \Big(\sum_x < \gamma_x >< \boldsymbol{u}_x \boldsymbol{u}_x^T > + < \gamma_\alpha > I_K \Big)^{-1}, \\
\boldsymbol{W} &= \Sigma_{\boldsymbol{\alpha}} \big(\sum_{s,x} < \gamma_x > \tilde{f}_{sx} < \boldsymbol{u}_x > \boldsymbol{p}_s^T \big) \big(\sum_s \boldsymbol{p}_s \boldsymbol{p}_s^T \big)^{-1}.
\end{aligned} \tag{7.17}$$

7.2.5.3 *Bayesian POD with neural networks (BayPOD-NN)*

The linearity assumption for the mean in (7.7) is potentially limiting the model expressiveness. Therefore, as a more flexible model, we let $\boldsymbol{\mu}_{\boldsymbol{w}}(\cdot)$ and $\Sigma_{\boldsymbol{w}}(\cdot)$, i.e. the mean and covariance matrix mapping, take the form of a neural network with weights $\boldsymbol{w}$ and input parameters $\boldsymbol{p}$. Neural networks have been widely used as the inference model in amortized variational inference [195, 196, 118, 197]. We denote this model with **BayPOD-NN**. Note that here, both the mean and covariance matrix are flexible functions of the input parameters. For BayPOD-NN, the corresponding inference procedure adopts the following amortized variational inference to update $\boldsymbol{\alpha}$:

Update $\boldsymbol{\alpha}$: To update the parameters of the neural network, which outputs the variational distribution over $\boldsymbol{\alpha}$, we adopt the stochastic gradient variational Bayes (SGVB) [195] algorithm to optimize (7.14).

The parameters of the mapping from the inputs to the variational distribution (i.e. $\boldsymbol{w}$) and the other variational parameters are alternately updated in the inference procedure.

7.3 Results and Discussion

7.3.1 Heated Rod Example

The first case study considers predicting the evolution of the temperature in a one-dimensional heated rod given time-dependent boundary conditions. Our output quantity of interest is the discretized temperature distribution along a rod of length L. Having the initial conditions and the specified boundary conditions, the evolution of the temperature field over the rod is governed by the heat equation with the diffusivity parameter, κ as an input. The temperature field varies as a function of time and distance along the rod. The initial condition and Dirichlet boundary conditions are defined as $f(\boldsymbol{x} = 0, t) = 3\sin(2t)$, $f(\boldsymbol{x} = L, t) = 3$, $f(\boldsymbol{x}, t = 0) = 0$, where we have a time-varying boundary condition at the left end of the rod and a fixed temperature value at the right end.

The input parameter vector for this example is $\boldsymbol{p} = [t, \kappa]$, and each snapshot is a discretized temperature field with $n_x = 200$ that corresponds to a set of values for the input vector, i.e. a fixed

diffusivity and time point. Each entry in a snapshot vector represents a spatial location along the rod.

Similar as in [190], we incorporate the constraints to satisfy the boundary conditions through a particular solution $\bar{\boldsymbol{f}}$ as in (7.2). For this, we can solve two auxiliary problems, one with boundary conditions $f(\boldsymbol{x} = 0, t) = 0$, $f(\boldsymbol{x} = L, t) = 1$ to get the steady-state solution $\bar{f}_L(\boldsymbol{x})$, and the other with boundary conditions $f(\boldsymbol{x} = 0, t) = 1$, $f(\boldsymbol{x} = L, t) = 0$ to get the steady-state solution $\bar{f}_0(\boldsymbol{x})$. The particular solution can then be defined as

$$\bar{\boldsymbol{f}} = 3\sin(2t)\bar{\boldsymbol{f}}_0(\boldsymbol{x}) + 3\bar{\boldsymbol{f}}_L(\boldsymbol{x}). \tag{7.18}$$

By subtracting the particular solution that corresponds to a snapshot from it, we get a modified snapshot with homogeneous boundary conditions. The different POD learning methods are then applied to the modified (training) snapshots to learn the reduced-order models and predict the temperature field for the unseen (test) snapshots satisfying the homogeneous boundary conditions. Adding the corresponding particular solution to each snapshot prediction guarantees satisfying the original inhomogeneous boundary conditions.

Snapshots are generated for six different diffusivity parameters κ =[0.25, 0.35, 0.45, 0.55, 0.65, 0.75]. For solving the heat equation, 628 equally spaced temporal points in $[0, 2\pi]$ are used, and 157 time points are randomly selected for snapshot generation. Overall, we have 942 snapshots corresponding to the 6 diffusivity values at 157 different time points.

For evaluating the different methods, 31 of the generated snapshots are randomly selected and withheld from the different methods for POD learning as the test set. The methods are trained on the remaining snapshots and make predictions for the withheld test snapshots. We set the dimension of POD bases, K, to be 5, which is the lowest number that results in less than 1% reconstruction error of the training snapshots by the classical POD analysis.

Method	mean	std	min	max
Polynomial Regression	0.846	0.370	0.213	1.685
BayPOD-LM	0.847	0.367	0.198	1.687
Neural Net Regression	0.041	0.019	0.004	0.079
BayPOD-NN	0.030	0.017	0.003	0.067

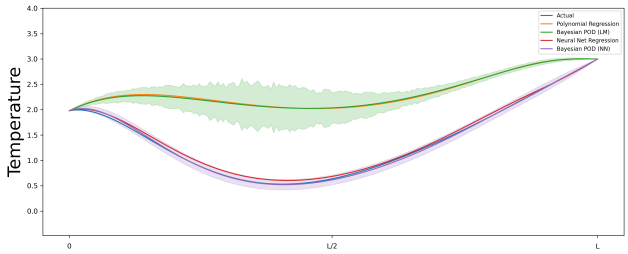
Table 7.1: Mean, standard deviation, minimum, and maximum of mean absolute error for Polynomial Regression, BayPOD-LM, Neural Network Regression, and BayPOD-NN on the different test snapshots for the heated rod case study.

7.3.1.1 Results

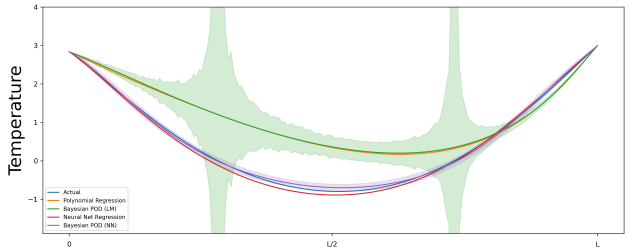
We test the performance of the proposed BayPOD-LM and BayPOD-NN for this case study and compare them with the original two-step approach using the corresponding polynomial regression (quadratic) and neural network regression (NN Regression) in [190]. In BayPOD-LM, the same polynomial features that are utilized in the original two-step Polynomial Regression approach are used to have quadratic regression and the variational parameters are initialized with the corresponding parameters from the original approach. Both BayPOD-NN and the two-step NN Regression employ the same NN architecture, having two hidden layers each with 50 nodes and ReLU activation functions. BayPOD-NN has outputs with the softplus activation for the covariance of $\boldsymbol{\alpha}$ ($\Sigma_{\boldsymbol{w}}(\cdot)$) in addition to the outputs for the mean of $\boldsymbol{\alpha}$ ($\boldsymbol{\mu}_{\boldsymbol{w}}(\cdot)$) for uncertainty quantification.

We calculate the mean absolute prediction error of each method for each test snapshot. The mean, standard deviation, minimum, and maximum values of the mean absolute errors of each method are provided in Table 7.1. Moreover, four test snapshots as representatives of the different patterns observed in all the test snapshots, their corresponding predictions by all the methods, and uncertainty estimates (as shaded regions) from the proposed BayPOD methods are shown in Figure 7.2.

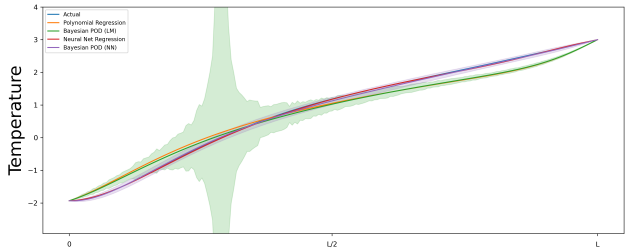
BayPOD-LM and the two-step Polynomial Regression have virtually the same error statistics as shown in Table 7.1. In Figure 7.2, we can see that models with the quadratic mapping from the inputs to the projection coefficients, i.e. Polynomial Regression and BayPOD-LM, have a relatively higher error compared with methods with a more flexible mapping using neural networks.



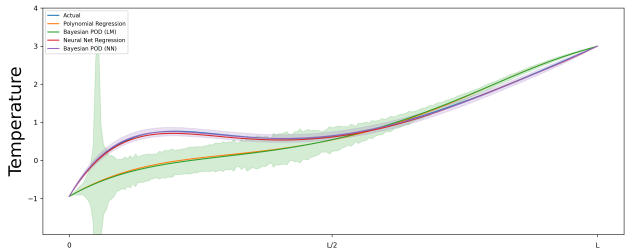
(a)



(b)



(c)



(d)

Figure 7.2: Four examples of comparing the actual temperature field and predictions from Polynomial Regression, BayPOD-LM, Neural Network Regression, and BayPOD-NN.

One of the critical advantages of BayPOD-LM is its uncertainty quantification capability compared with the two-step Polynomial Regression. The estimated 95% posterior confidence intervals by BayPOD-LM in Figures 7.2(c)-(d) include the true temperature values across many spatial points. Moreover, in Figure 7.2(a), although both BayPOD-LM and Polynomial Regression have very large errors and the 95% posterior confidence interval from BayPOD-LM is far from the true values, we can see that the prediction uncertainty is highly correlated with the prediction error over the length of the rod.

Table 7.1 and Figure 7.2 clearly show the advantage of the more flexible BayPOD-NN modeling compared with BayPOD-LM. It is clear that BayPOD-NN has significantly more accurate predictions in addition to narrower and better uncertainty estimates. Both BayPOD-NN and NN Regression outperform the corresponding linear models. From Table 7.1, we see that BayPOD-NN also performs better than the two-step NN Regression in terms of the error statistics while providing uncertainty quantification as shown in Figure 7.2. We can see in Figures 7.2(a)-(d) that the estimated 95% posterior confidence intervals by BayPOD-NN contain the true temperature values in all the spatial locations for the four depicted snapshots. The results of this case study clearly show that BayPOD-NN is more accurate than the deterministic two-step NN regression while providing principled and accurate uncertainty estimates.

7.3.2 Airfoil Example

This case study considers the prediction of the flow around an airfoil, using data generated from a large-scale computational fluid dynamics (CFD) simulation [199].

The input parameters for this example are the freestream Mach number, M, and the airfoil lift coefficient, c_l. Our input parameter vector is $\boldsymbol{p} = [M, c_l] \in \mathbb{R}^2$. The output quantity of interest is the pressure field around the airfoil, which varies as a function of the input parameters. In this example, we use the `SU2` CFD tool suite, a multi-purpose open-source solver, specifically developed for aerospace applications. `SU2` uses a finite volume method to discretize the underlying partial differential equations. Here we use the Euler equations to model the inviscid steady flow over the airfoil. We consider a range of Mach numbers, spanning subsonic and transonic flow

regimes. Flow tangency boundary conditions are imposed on the airfoil surface and the farfield boundary is approximately 20 chord lengths away from the airfoil.

`SU2`'s discretization of the pressure field has $n_x = 9027$ degrees of freedom; that is, each SU2 pressure field solution is a vector of dimension $n_x = 9027$, where each entry corresponds to the predicted pressure at a different spatial location in the computational domain.

We refer to each pressure field solution vector as a snapshot. Snapshots are generated for a domain of Mach numbers from $M = 0.6$ to $M = 0.8$ in increments of 0.01. At each Mach number, the following seven lift coefficients are used: $c_l = [0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9]$. This provides a total of $n_s = 147$ snapshots, where each snapshot is a high-fidelity pressure field solution, represented as a high-dimensional vector.

For evaluation, we withhold all data corresponding to a single Mach number, train the models with the remaining data, and test on the withheld data one sample at a time. We set the dimension of POD bases, K, to be 20, which is the lowest number that results in less than 1% reconstruction error of the training snapshots by the classical POD analysis for all the data splits.

7.3.2.1 Results of BayPOD-LM and discussion

Figure 7.3 illustrates the comparison of BayPOD-LM after five iterations of updates with the original results using polynomial regression (quadratic) in [190]. In this figure, for each Mach number, the plots show the minimum, mean and maximum of the mean absolute error (MAE) over the entire field for each of the seven lift coefficient values. Note that in BayPOD-LM, we use the same polynomial combinations of the features with degree less than or equal to 2, and initialize the variational parameters with the corresponding parameters from the original two-step approach. We can see from the figure that BayPOD-LM has a similar or slightly better performance compared with the two-step Polynomial Regression for the different Mach numbers.

For the case when the Mach number $M = 0.7$ and lift coefficient $c_l = 0.7$, Figure 7.4 shows the point-wise absolute error of the field predictions. All pressure fields produced with Mach 0.7 have been held out of the training set used by each method for making the predictions. This figure again shows that BayPOD-LM performs slightly better in terms of MAE. Critically, the

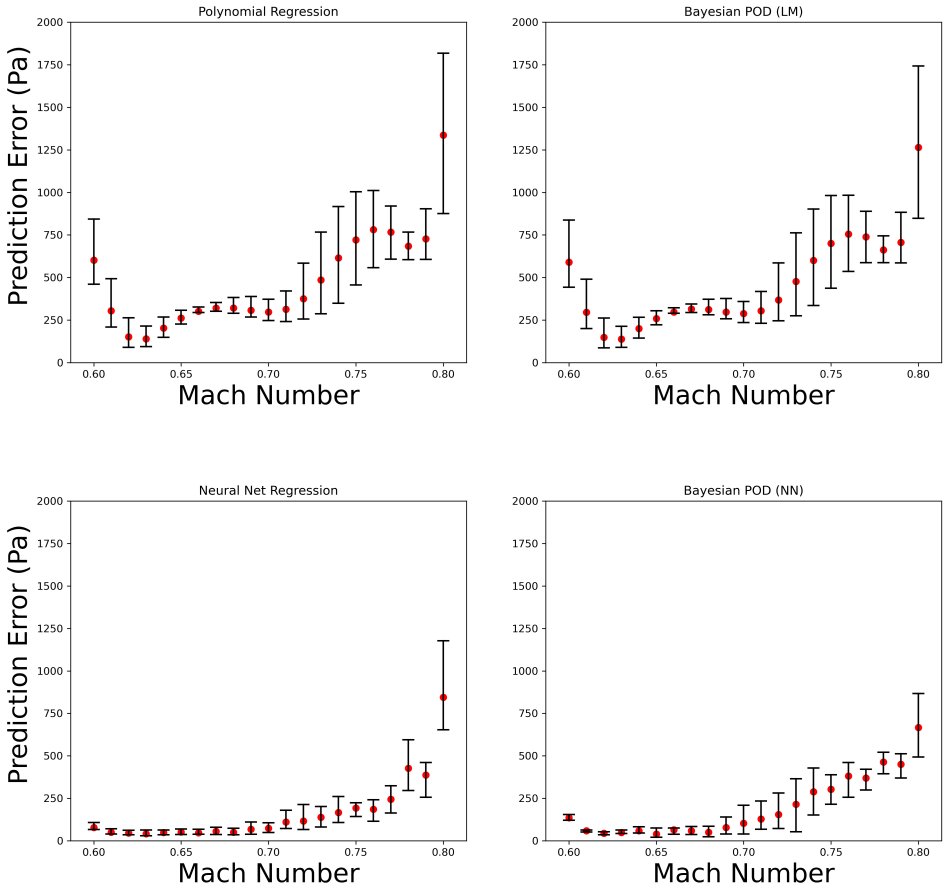
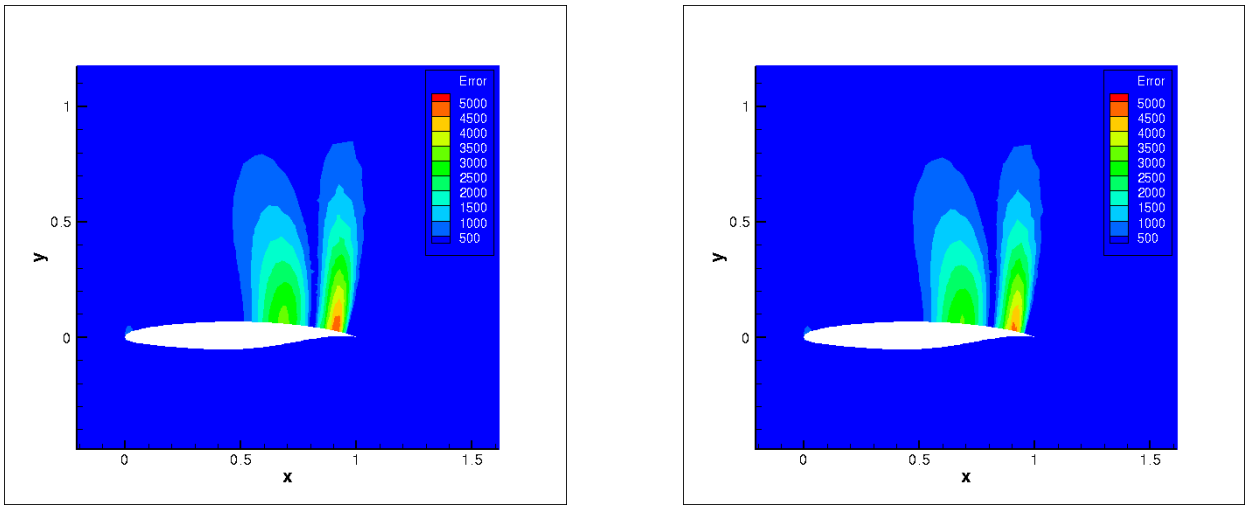


Figure 7.3: The minimum, mean, and maximum mean absolute error for (a) Polynomial Regression in top left, (b) BayPOD-LM in top right, (c) Neural Network Regression in bottom left, and (d) BayPOD-NN in bottom right.

advantage of BayPOD-LM is its uncertainty quantification capability. In Figure 7.5, the point-wise posterior predictive standard deviation is illustrated, where the regions with higher uncertainty are overlapping with some of the regions with the highest MAE in Figure 7.4, demonstrating the effectiveness of the uncertainty quantification capability of BayPOD.

7.3.2.2 Results of BayPOD-NN and discussion

Next, we test the more flexible BayPOD-NN with the same setup of this case study. We use a NN with two hidden layers, each with 50 nodes and ReLU activation functions, shared by both $\boldsymbol{\mu}_{\boldsymbol{w}}(\cdot)$ and $\Sigma_{\boldsymbol{w}}(\cdot)$. The output layer for the mean inference network does not have any activation



(a) Polynomial Regression

(b) BayPOD-LM

Figure 7.4: The error field produced by predictions

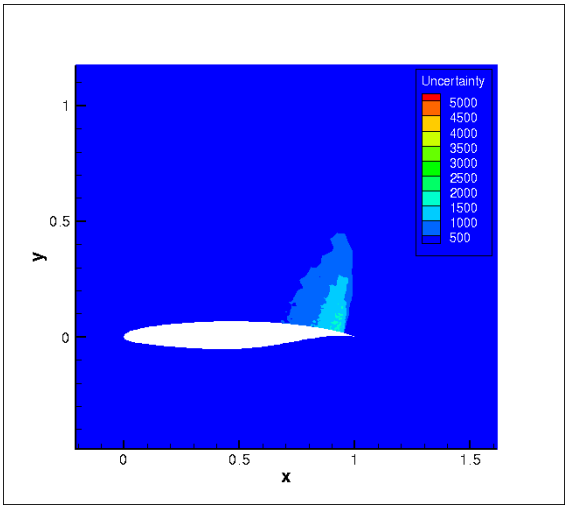
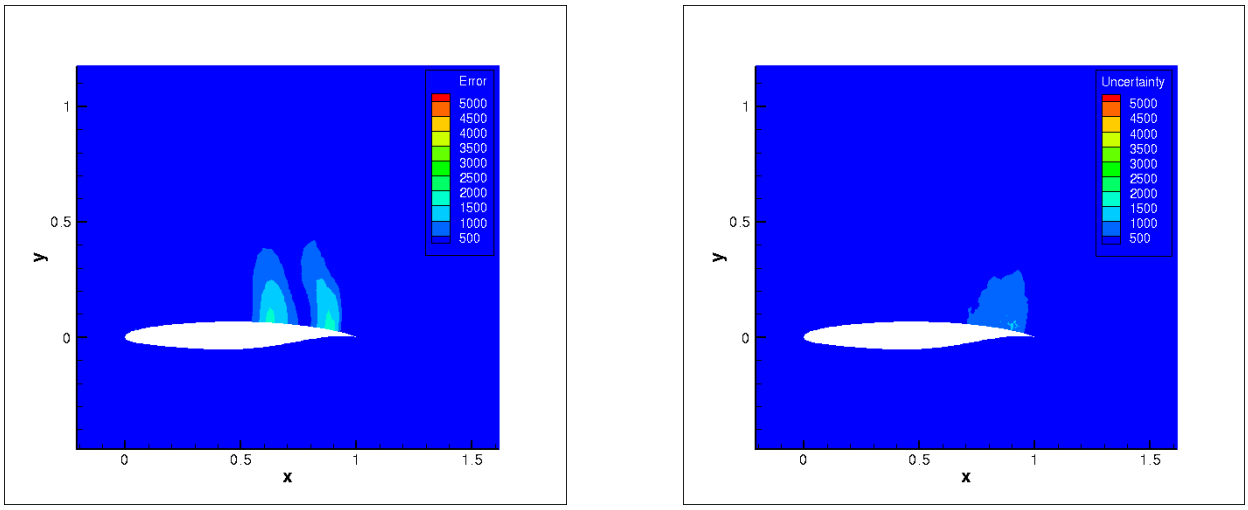


Figure 7.5: The posterior predictive standard deviation from BayPOD-LM

while we use softplus for the covariance network.

For direct comparison, we here illustrate the results with the same Mach number $M = 0.7$ and lift coefficient $c_l = 0.7$. The point-wise absolute error of the field predictions by BayPOD-NN and the posterior predictive standard deviation as a measure of uncertainty are provided in Figure 7.6. We can clearly see that BayPOD-NN improves upon BayPOD-LM.



(a) Error field

(b) Uncertainty

Figure 7.6: The prediction from BayPOD-NN

Moreover, we compare BayPOD-NN with the original Polynomial Regression approach and also Neural Network Regression (NN Regression) in terms of the error statistics over the entire field for the lift coefficients and the different Mach numbers, as shown in Figure 7.3(d). The figure depicts the advantage of the more flexible BayPOD-NN modeling compared with the linear model in BayPOD-LM in Figure 7.3(b), where we see consistent improvement for all Mach numbers.

In Figure 7.3(c) and (d), the two-step NN Regression approach and BayPOD-NN overall show comparable performance in terms of the error statistics over the entire field for the lift coefficients and different Mach numbers. The NN architecture is the same for both NN Regression and BayPOO-NN (i.e. two hidden layers with width 50), with BayPOD-NN having the additional outputs corresponding to the covariance of $\boldsymbol{\alpha}$ $(\Sigma_{\boldsymbol{w}}(\cdot))$. We can see that for smaller Mach numbers their error statistics are virtually the same, while for a few of the mid-range Mach numbers NN Regression has slightly better error statistics and for larger Mach numbers BayPOD-NN performs better. These results clearly show that BayPOD-NN is a flexible unified approach for learning projection bases and coefficients that does not lose accuracy compared with the deterministic two-step NN Regression, while providing a mechanism for principled input-dependent uncertainty estimates.

It is worth emphasizing that as opposed to the NN Regression approach where NNs are used as the regressor in the two-step procedure, for BayPOD-NN, we are integrating NNs in the variational distribution, where in addition to providing uncertainty estimates, the structure of the model and the prior distributions automatically impose regularizations obviating the need for additional fine-tuned regularization.

8. SUMMARY AND CONCLUSION

In this dissertation we have developed methods and frameworks to integrate existing prior knowledge and data from other domains to design predictors with improved accuracy, reliability, and uncertainty estimation in the target domain and the application of interest.

In Chapter 2, we have proposed a knowledge-driven prior construction method with a general framework of constraints. We have shown how prior biological knowledge can be mapped into a set of constraints. Knowledge can come from biological signaling pathways and other population studies, and be translated into constraints over conditional probabilities. The superior performance of this general scheme is shown on two important pathway families, the mammalian cell-cycle pathway and the pathway centering around TP53. In addition, prior construction and the optimal Bayesian classifier (OBC) are extended to a mixture model, where data sets are with missing labels. Moreover, comparisons on a publicly available gene expression dataset show that classification performance can be significantly improved for small sample sizes when corresponding pathway prior knowledge is integrated for constructing prior probabilities. Prior construction is extended to regression and Gaussian mixture models in Chapter 3 which is useful for modeling data heterogeneity.

We have proposed a new Bayesian domain adaptation framework for leveraging labeled data from other domains for next-generation sequencing (NGS) count data in Chapter 4, and developed optimal Bayesian supervised domain adaptation (OBSDA) with an efficient Gibbs sampler. Compared to existing methods for domain adaptation and transfer learning, OBSDA has the following features: It uses label information across domains for transfer learning compared with unsupervised models. It models the relationship between different domains as well as different classes in one domain, contrasting with existing supervised methods that are restricted to the cases requiring domains having the same labels. It can leverage data from domains containing no common labels with no negative effect on the learning task for the target domain. In addition, when analyzing NGS data, it does not need any ad-hoc normalization of the counts due to its generative nature.

Moreover, we have introduced an extension of OBSDA, SI-OBSDA, where flexible variational distributions are formed by using neural networks as an implicit generator. We have proposed incorporating prior knowledge in terms of gene-gene network connectivity as constraints imposed on the latent embedding to construct informed approximate posteriors to improve the performance. Our experiments on the real-world RNA-Seq data show that by sharing information across domains and labels, OBSDA achieves the best cancer subtype identification performance compared with methods using only target domain data and other methods that try to use all the domains' data. Additionally, the results show that by incorporating the prior knowledge, SI-OBSDA can further improve the subtype identification accuracy.

In Chapter 5, we have addressed the problem of clustering with missing values in smaller sample sizes, where we have incorporated the generation of missing values with the original generating random labeled point process. We have derived the optimal clusterer for different scenarios in which features are distributed according to multivariate Gaussian distributions, and have verified the superior performance of the proposed method in both simulations and a real-world application with missing data. In this Chapter, we have, in effect, confronted an old problem in signal processing: If we wish to make a decision based on a noisy observed signal, is it better to filter the observed signal and then determine the optimal decision on the filtered signal, or to find the optimal decision based directly on the observed signal? The answer is the latter. The reason is that the latter approach is fully optimal relative to the actual observation process, whereas, even if in the first approach the filtering is optimal relative to the noise process, the first approach produces a composite of two actions, filter and decision, each of which is only optimal relative to a portion of the actual observation process.

In the first part of Chapter 6 we have explained optimal experimental design and presented a generalized MOCU framework, leading to the MOCU-based experimental design pertaining to the maximum uncertainty reduction of differential cost with respect to the actual operational objectives. The proposed framework fits into classical Bayesian experimental design and is more flexible for the development of corresponding experimental design strategies for different real-

world applications compared to the existing methods with their corresponding model assumptions. Our generalized MOCU framework, with the benefits from flexible dissection of the uncertainty class, action (operator) space, experiment space, and utility function depending on operational objectives, can lead to better objective-based uncertainty quantification and thereafter better experimental design to achieve desired objectives with smaller operational cost. In the second part of the Chapter, we have developed an efficient Bayesian experiment design framework under model uncertainty that can leverage prior knowledge regarding the potential models, and have successfully applied it to materials discovery in single- and multi-objective material property space using a test set of MAX phases with promising results.

Finally, we have developed a new framework of Bayesian reduced-order models in Chapter 7, that enable reliable estimates of prediction confidence for robust decision making in addition to incorporating the underlying scientific principles or physics constraints (i.e., the existing prior knowledge). One critical contribution of developing learnable Bayesian reduced-order models is to not only seek the best low-dimensional subspace to approximate high-dimensional dynamics, but also allow uncertainty estimates by learning distributions of reduced order model parameters. By modeling both projections and mappings from system inputs to projection coefficients in one unified model with seamless integration of inference for both components, our experimental results with the heated rod and airfoil examples clearly show the advantages over non-Bayesian reduced-order models on both prediction accuracy and uncertainty estimation of high-dimensional system dynamics. With the developed BayPOD framework, Bayesian experimental design guided by efficient predictive models constrained by scientific principles can be developed for science and engineering applications where training data are difficult or costly to generate and the involved decision making based on predictions can have significant consequences.

REFERENCES

[1] L. West, S. J. Vidwans, N. P. Campbell, J. Shrager, G. R. Simon, R. Bueno, P. A. Dennis, G. A. Otterson, and R. Salgia, "A novel classification of lung cancer into molecular subtypes," *PLOS ONE*, vol. 7, pp. 1–11, 02 2012.

[2] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, p. e124, 2006.

[3] R. K. Layek, A. Datta, and E. R. Dougherty, "From biological pathways to regulatory networks," *Mol. BioSyst.*, vol. 7, pp. 843–851, 2011.

[4] A. A. Alizadeh, V. Aranda, A. Bardelli, C. Blanpain, C. Bock, C. Borowski, C. Caldas, A. Califano, M. Doherty, M. Elsner, *et al.*, "Toward understanding and exploiting tumor heterogeneity," *Nature medicine*, vol. 21, no. 8, pp. 846–853, 2015.

[5] E. R. Dougherty, A. Zollanvari, and U. M. Braga-Neto, "The illusion of distribution-free small-sample classification in genomics," *Current Genomics*, vol. 12, no. 5, p. 333, 2011.

[6] E. R. Dougherty and L. A. Dalton, "Scientific knowledge is possible with small-sample classification," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2013, no. 1, pp. 1–12, 2013.

[7] E. T. Jaynes, "What is the question?," in *Bayesian statistics* (J. Bernardo, M. deGroot, D. Lindly, and A. Smith, eds.), Valencia: Valencia Univ. Press, 1980.

[8] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[9] A. Zellner, *Past and Recent Results on Maximal Data Information Priors*. Working paper series in economics and econometrics, Chicago: University of Chicago, Graduate School of

Business, Department of Economics, 1995.

[10] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[11] C. Rodríguez, "Entropic priors," tech. rep., Department of Mathematics and Statistics, State University of New York, Albany, 1991.

[12] J. Berger and J. Bernardo, "On the development of reference priors," *Bayesian statistics*, vol. 4, no. 4, pp. 35–60, 1992.

[13] J. Spall and S. Hill, "Least-informative Bayesian prior distributions for finite samples based on information theory," *IEEE Transactions on Automatic Control*, vol. 35, no. 5, pp. 580–583, 1990.

[14] J. Bernardo, "Reference posterior distributions for Bayesian inference," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 2, pp. 113–147, 1979.

[15] R. Kass and L. Wasserman, "The selection of prior distributions by formal rules," *Journal of the American Statistical Association*, vol. 91, no. 435, pp. 1343–1370, 1996.

[16] J. Berger, J. Bernardo, and D. Sun, "Objective priors for discrete parameter spaces," *Journal of the American Statistical Association*, vol. 107, no. 498, pp. 636–648, 2012.

[17] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[18] E. T. Jaynes, "Prior probabilities," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 3, pp. 227–241, 1968.

[19] A. Zellner, "Models, prior information, and Bayesian analysis," *Journal of Econometrics*, vol. 75, no. 1, pp. 51–68, 1996.

[20] J. Burg, D. Luenberger, and D. Wenger, "Estimation of structured covariance matrices," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 963–974, 1982.

[21] K. Werner, M. Jansson, and P. Stoica, "On estimation of covariance matrices with kronecker product structure," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 478–491, 2008.

[22] A. Wiesel and A. Hero, "Distributed covariance estimation in Gaussian graphical models," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 211–220, 2011.

[23] A. Wiesel, Y. Eldar, and A. Hero, "Covariance estimation in decomposable Gaussian graphical models," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1482–1492, 2010.

[24] T. Breslin, M. Krogh, C. Peterson, and C. Troein, "Signal transduction pathway profiling of individual tumor samples," *BMC bioinformatics*, vol. 6, no. 1, p. 163, 2005.

[25] Y. Zhu, X. Shen, and W. Pan, "Network-based support vector machine for classification of microarray samples," *BMC Bioinformatics*, vol. 10, no. 1, p. S21, 2009.

[26] J. P. Svensson, L. J. Stalpers, R. E. Esveldt-van Lange, N. A. Franken, J. Haveman, B. Klein, I. Turesson, H. Vrieling, and M. Giphart-Gassler, "Analysis of gene expression using gene sets discriminates cancer patients with and without late radiation toxicity," *PLoS medicine*, vol. 3, no. 10, p. e422, 2006.

[27] E. Lee, H.-Y. Chuang, J.-W. Kim, T. Ideker, and D. Lee, "Inferring pathway activity toward precise disease classification," *PLoS computational biology*, vol. 4, no. 11, p. e1000217, 2008.

[28] J. Su, B.-J. Yoon, and E. R. Dougherty, "Accurate and reliable cancer classification based on probabilistic inference of pathway activity," *PLoS One*, vol. 4, no. 12, p. e8161, 2009.

[29] H.-S. Eo, J. Y. Heo, Y. Choi, Y. Hwang, and H.-S. Choi, "A pathway-based classification of breast cancer integrating data on differentially expressed genes, copy number variations and microrna target genes," *Molecules and Cells*, vol. 34, no. 4, pp. 393–398, 2012.

[30] Z. Wen, Z.-P. Liu, Y. Yan, G. Piao, Z. Liu, J. Wu, and L. Chen, "Identifying responsive modules by mathematical programming: An application to budding yeast cell cycle," *PloS one*, vol. 7, no. 7, p. e41854, 2012.

[31] S. Kim, M. Kon, C. DeLisi, *et al.*, "Pathway-based classification of cancer subtypes," *Biology direct*, vol. 7, no. 1, pp. 1–22, 2012.

[32] N. Khunlertgit and B.-J. Yoon, "Identification of robust pathway markers for cancer through rank-based pathway activity inference," *Advances in bioinformatics*, vol. 2013, pp. Article ID 618461–8, 2013.

[33] P. Wei and W. Pan, "Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model," *Bioinformatics*, vol. 24, no. 3, pp. 404–411, 2007.

[34] P. Wei and W. Pan, "Network-based genomic discovery: application and comparison of Markov random-field models," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 59, no. 1, pp. 105–125, 2010.

[35] P. Wei and W. Pan, "Bayesian joint modeling of multiple gene networks and diverse genomic data to identify target genes of a transcription factor," *The Annals of Applied Statistics*, vol. 6, no. 1, pp. 334–355, 2012.

[36] M. L. Gatza, J. E. Lucas, W. T. Barry, J. W. Kim, Q. Wang, M. D. Crawford, M. B. Datto, M. Kelley, B. Mathey-Prevot, A. Potti, *et al.*, "A pathway-based classification of human breast cancer," *Proceedings of the National Academy of Sciences*, vol. 107, no. 15, pp. 6994–6999, 2010.

[37] J. R. Nevins, "Pathway-based classification of lung cancer: a strategy to guide therapeutic selection," *Proceedings of the American Thoracic Society*, vol. 8, no. 2, p. 180, 2011.

[38] Z. Wen, Z.-P. Liu, Z. Liu, Y. Zhang, and L. Chen, "An integrated approach to identify causal network modules of complex diseases with application to colorectal cancer," *Journal of the American Medical Informatics Association*, vol. 20, no. 4, pp. 659–667, 2013.

[39] M. S. Esfahani and E. R. Dougherty, "Incorporation of biological pathway knowledge in the construction of priors for optimal Bayesian classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, pp. 202–218, Jan. 2014.

[40] M. S. Esfahani and E. R. Dougherty, "An optimization-based framework for the transformation of incomplete biological knowledge into a probabilistic structure and its application to the utilization of gene/protein signaling pathways in discrete phenotype classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 6, pp. 1304–1321, 2015.

[41] S. Guiasu and A. Shenitzer, "The principle of maximum entropy," *The Mathematical Intelligencer*, vol. 7, no. 1, pp. 42–48, 1985.

[42] J. Hua, C. Sima, M. Cypert, G. C. Gooden, S. Shack, L. Alla, E. A. Smith, J. M. Trent, E. R. Dougherty, and M. L. Bittner, "Tracking transcriptional activities with high-content epifluorescent imaging," *Journal of biomedical optics*, vol. 17, no. 4, pp. 0460081–04600815, 2012.

[43] L. Dalton and E. Dougherty, "Optimal classifiers with minimum expected error within a Bayesian framework–part I: Discrete and Gaussian models," *Pattern Recognition*, vol. 46, no. 5, pp. 1301–1314, 2013.

[44] L. Dalton and E. R. Dougherty, "Optimal classifiers with minimum expected error within a Bayesian framework–part II: Properties and performance analysis," *Pattern Recognition*, vol. 46, no. 5, pp. 1288–1300, 2013.

[45] L. Dalton and E. Dougherty, "Bayesian minimum mean-square error estimation for classification error–part I: Definition and the bayesian MMSE error estimator for discrete classification," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 115–129, 2011.

[46] D. J. C. MacKay, "Introduction to Monte Carlo methods," in *Learning in Graphical Models* (M. I. Jordan, ed.), NATO Science Series, pp. 175–204, Dordrecht, Netherlands: Kluwer Academic Press, 1998.

[47] G. Casella and E. I. George, "Explaining the Gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[48] C. P. Robert and G. Casella, *Monte carlo statistical methods*. New York: Springer Science + Business Media, 2004.

[49] A. Zellner, *Maximal Data Information Prior Distributions, Basic Issues in Econometrics*. The University of Chicago Press, Chicago, USA, 1984.

[50] N. Ebrahimi, E. Maasoumi, and E. S. Soofi, *Measuring Informativeness of Data by Entropy and Variance*, pp. 61–77. Heidelberg: Physica-Verlag HD, 1999.

[51] E. R. Dougherty, M. Brun, J. M. Trent, and M. L. Bittner, "Conditioning-based modeling of contextual genomic regulation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 2, pp. 310–320, 2009.

[52] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437 – 467, 1969.

[53] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, p. 261, 2002.

[54] R. Weinberg, *The biology of cancer*. New York: Garland science, 2013.

[55] M. S. Esfahani, B.-J. Yoon, and E. R. Dougherty, "Probabilistic reconstruction of the tumor progression process in gene regulatory networks in the presence of uncertainty," *BMC Bioinformatics*, vol. 12, no. 10, p. S9, 2011.

[56] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and regression trees*. Boca Raton: Chapman & Hall/CRC, 1984.

[57] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[59] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. Cambridge, MA, USA: MIT Press, 2001.

[60] American Cancer Society, Atlanta, *Cancer Facts and Figures 2017*, 2017.

[61] J. Gao, B. A. Aksoy, U. Dogrusoz, G. Dresdner, B. Gross, S. O. Sumer, Y. Sun, A. Jacobsen, R. Sinha, E. Larsson, E. Cerami, C. Sander, and N. Schultz, "Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal," *Science Signaling*, vol. 6, no. 269, pp. pl1–pl1, 2013.

[62] E. Cerami, J. Gao, U. Dogrusoz, B. E. Gross, S. O. Sumer, B. A. Aksoy, A. Jacobsen, C. J. Byrne, M. L. Heuer, E. Larsson, Y. Antipin, B. Reva, A. P. Goldberg, C. Sander, and N. Schultz, "The cBio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data," *Cancer Discovery*, vol. 2, no. 5, pp. 401–404, 2012.

[63] M. S. Lawrence, P. Stojanov, P. Polak, G. V. Kryukov, K. Cibulskis, A. Sivachenko, S. L. Carter, C. Stewart, C. H. Mermel, S. A. Roberts, *et al.*, "Mutational heterogeneity in cancer and the search for new cancer-associated genes," *Nature*, vol. 499, no. 7457, pp. 214–218, 2013.

[64] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[65] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "KEGG as a reference resource for gene and protein annotation," *Nucleic acids research*, vol. 44, no. D1, pp. D457–D462, 2016.

[66] J. Lortet-Tieulent, I. Soerjomataram, J. Ferlay, M. Rutherford, E. Weiderpass, and F. Bray, "International trends in lung cancer incidence by histological subtype: Adenocarcinoma stabilizing in men but still increasing in women," *Lung Cancer*, vol. 84, no. 1, pp. 13 – 22, 2014.

[67] S. Richardson and P. J. Green, "On Bayesian analysis of mixtures with an unknown number of components (with discussion)," *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 59, no. 4, pp. 731–792, 1997.

[68] M. West and M. D. Escobar, *Hierarchical priors and mixture models, with application in regression and density estimation*. Institute of Statistics and Decision Sciences, Duke University, 1993.

[69] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis / K.V. Mardia, J.T. Kent, J.M. Bibby*. Academic Press London ; New York, 1979.

[70] H. G. Sung, *Gaussian mixture regression and classification*. PhD thesis, Rice University, 2004.

[71] X. Qian and E. Dougherty, "Bayesian regression with network prior: Optimal Bayesian filtering perspective," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6243–6253, 2016.

[72] L. A. Dalton and M. R. Yousefi, "On optimal bayesian classification and risk estimation under multiple classes," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2015, no. 1, p. 8, 2015.

[73] L. A. Dalton and E. R. Dougherty, "Intrinsically optimal Bayesian robust filtering," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 657–670, 2014.

[74] H. Akaike, "A bayesian analysis of the minimum aic procedure," *Annals of the Institute of Statistical mathematics*, vol. 30, no. 1, pp. 9–14, 1978.

[75] H. Bozdogan, "Model selection and akaike's information criterion (aic): The general theory and its analytical extensions," *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[76] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[77] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the em algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.

[78] J. Diebolt and C. P. Robert, "Estimation of finite mixture distributions through Bayesian sampling," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 363–375, 1994.

[79] L. Wasserman, "Asymptotic inference for mixture models by using data-dependent priors," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 1, pp. 159–180, 2000.

[80] J.-M. Marin, K. L. Mengersen, and C. Robert, "Bayesian modelling and inference on mixtures of distributions," in *Handbook of Statistics: Volume 25* (D. Dey and C. Rao, eds.), Elsevier, 2005.

[81] A. Jasra, C. C. Holmes, and D. A. Stephens, "Markov chain monte carlo methods and the label switching problem in Bayesian mixture modeling," *Statist. Sci.*, vol. 20, pp. 50–67, 02 2005.

[82] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–1182, 2008.

[83] M. L. Garcia-Vaquero, M. Gama-Carvalho, J. De Las Rivas, and F. R. Pinto, "Searching the overlap between network modules with specific betweeness (S2B) and its application to cross-disease analysis," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[84] M. Gustafsson, C. E. Nestor, H. Zhang, A.-L. Barabási, S. Baranzini, S. Brunak, K. F. Chung, H. J. Federoff, A.-C. Gavin, R. R. Meehan, *et al.*, "Modules, networks and systems medicine for understanding disease and aiding diagnosis," *Genome medicine*, vol. 6, no. 10, pp. 1–11, 2014.

[85] J. Menche, A. Sharma, M. Kitsak, S. D. Ghiassian, M. Vidal, J. Loscalzo, and A.-L. Barabási, "Uncovering disease-disease relationships through the incomplete interactome," *Science*, vol. 347, no. 6224, 2015.

[86] D. A. Levine, "Integrated genomic characterization of endometrial carcinoma," *Nature*, vol. 497, no. 7447, pp. 67–73, 2013.

[87] P. Wei and W. Pan, "Bayesian joint modeling of multiple gene networks and diverse genomic data to identify target genes of a transcription factor," *The Annals of Applied Statistics*, vol. 6, no. 1, pp. 334 – 355, 2012.

[88] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Incorporating biological prior knowledge for Bayesian learning via maximal knowledge-driven information priors," *BMC bioinformatics*, vol. 18, no. 14, p. 552, 2017.

[89] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Constructing pathway-based priors within a Gaussian mixture model for Bayesian regression and classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 524–537, 2017.

[90] C. Stark, B.-J. Breitkreutz, A. Chatr-Aryamontri, L. Boucher, R. Oughtred, M. S. Livstone, J. Nixon, K. Van Auken, X. Wang, X. Shi, *et al.*, "The BioGRID interaction database: 2011 update," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D698–D704, 2010.

[91] B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, M. Feuermann, A. Ghanbarian, S. Kerrien, J. Khadake, *et al.*, "The IntAct molecular interaction database in 2010," *Nucleic acids research*, vol. 38, no. suppl_1, pp. D525–D531, 2010.

[92] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[93] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[94] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[95] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, pp. 193–200, 2007.

[96] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[97] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[98] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *International Conference on Machine Learning*, pp. 222–230, 2013.

[99] L. Jacob, J.-p. Vert, and F. R. Bach, "Clustered multi-task learning: A convex formulation," in *Advances in neural information processing systems*, pp. 745–752, 2009.

[100] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 521–528, 2011.

[101] A. Passos, P. Rai, J. Wainer, and H. Daumé III, "Flexible modeling of latent task structures in multitask learning," in *Proceedings of the 29th International Coference on Machine Learning*, pp. 1283–1290, 2012.

[102] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[103] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, pp. 97–105, 2015.

[104] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, pp. 469–477, 2016.

[105] R. Normand, W. Du, M. Briller, R. Gaujoux, E. Starosvetsky, A. Ziv-Kenet, G. Shalev-Malul, R. J. Tibshirani, and S. S. Shen-Orr, "Found in translation: a machine learning model for mouse-to-human inference," *Nature methods*, vol. 15, no. 12, pp. 1067–1073, 2018.

[106] S. R. Dhruba, R. Rahman, K. Matlock, S. Ghosh, and R. Pal, "Application of transfer learning for cancer drug sensitivity prediction," *BMC bioinformatics*, vol. 19, no. 17, p. 497, 2018.

[107] E. Hajiramezanali, S. Z. Dadaneh, A. Karbalayghareh, M. Zhou, and X. Qian, "Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data," in *Advances in Neural Information Processing Systems 31*, pp. 9115–9124, 2018.

[108] A. Karbalayghareh, X. Qian, and E. R. Dougherty, "Optimal Bayesian transfer learning for count data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 2, pp. 644–655, 2021.

[109] L. A. Dalton and E. R. Dougherty, *Optimal Bayesian Classification*. SPIE Press, 2020.

[110] M. Zhou and L. Carin, "Negative binomial process count and mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 307–320, Feb 2015.

[111] M. Yin and M. Zhou, "Semi-implicit variational inference," in *International Conference on Machine Learning*, pp. 5660–5669, PMLR, 2018.

[112] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2010.

[113] S. Z. Dadaneh, X. Qian, and M. Zhou, "Bnp-seq: Bayesian nonparametric differential expression analysis of sequencing count data," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 81–94, 2018.

[114] P. Rai and H. Daumé, "The infinite hierarchical factor regression model," in *Advances in Neural Information Processing Systems*, pp. 1321–1328, 2009.

[115] M. Zhou, "Nonparametric Bayesian negative binomial factor analysis," *Bayesian Analysis*, vol. 13, no. 4, pp. 1065–1093, 2018.

[116] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, Nov 1999.

[117] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[118] S. Zamani Dadaneh, S. Boluki, M. Yin, M. Zhou, and X. Qian, "Pairwise supervised hashing with Bernoulli variational auto-encoder and self-control gradient estimator," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 540–549, PMLR, 2020.

[119] A. Karbalayghareh, X. Qian, and E. R. Dougherty, "Optimal bayesian transfer learning," *IEEE Transactions on Signal Processing*, vol. 66, no. 14, pp. 3724–3739, 2018.

[120] C. Hutter and J. C. Zenklusen, "The Cancer Genome Atlas: Creating lasting value beyond its data," *Cell*, vol. 173, no. 2, pp. 283–285, 2018.

[121] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, vol. 15, p. 550, Dec 2014.

[122] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[123] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[124] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 281–297, 1999. PMID: 10582567.

[125] M. Bittner, P. Meitzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Journal of Computational Biology*, vol. 406, no. 3, pp. 536–540, 2000.

[126] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.

[127] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[128] 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.

[129] S. Boluki, X. Qian, and E. R. Dougherty, "Experimental design via generalized mean objective cost of uncertainty," *IEEE Access*, vol. 7, pp. 2223–2230, 2018.

[130] A. Broumand, M. S. Esfahani, B.-J. Yoon, and E. R. Dougherty, "Discrete optimal Bayesian classification with error-conditioned sequential sampling," *Pattern Recognition*, vol. 48, no. 11, pp. 3766 – 3782, 2015.

[131] E. R. Dougherty and M. Brun, "A probabilistic theory of clustering," *Pattern Recognition*, vol. 37, no. 5, pp. 917 – 925, 2004.

[132] L. A. Dalton, M. E. Benalcázar, M. Brun, and E. R. Dougherty, "Analytic representation of Bayes labeling and Bayes clustering operators for random labeled point processes," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1605–1620, March 2015.

[133] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[134] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-seq," *Nature Methods*, vol. 5, no. 7, p. 621, 2008.

[135] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[136] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of Statistical Software*, pp. 1–68, 2010.

[137] J. Honaker, G. King, M. Blackwell, *et al.*, "Amelia ii: A program for missing data," *Journal of Statistical Software*, vol. 45, no. 7, pp. 1–47, 2011.

[138] D. J. Stekhoven and P. Bühlmann, "Missforest—non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2011.

[139] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 333. New Jersey: John Wiley & Sons, 2014.

[140] J. T. Chi, E. C. Chi, and R. G. Baraniuk, "k-pod: A method for k-means clustering of missing data," *The American Statistician*, vol. 70, no. 1, pp. 91–99, 2016.

[141] R. J. Hathaway and J. C. Bezdek, "Fuzzy c-means clustering of incomplete data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, pp. 735–744, Oct 2001.

[142] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[143] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[144] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[145] S. Z. Dadaneh, E. R. Dougherty, and X. Qian, "Optimal Bayesian classification with missing values," *IEEE Transactions on Signal Processing*, vol. 66, no. 16, pp. 4182–4192, 2018.

[146] S. Boluki, S. Z. Dadaneh, X. Qian, and E. R. Dougherty, "Optimal clustering with missing values," *BMC bioinformatics*, vol. 20, no. 12, p. 321, 2019.

[147] The Cancer Genome Atlas Research Network (TCGA), "Comprehensive genomic characterization defines human glioblastoma genes and core pathways," *Nature*, vol. 455, no. 7216, p. 1061, 2008.

[148] Y.-W. Wan, G. I. Allen, and Z. Liu, "TCGA2STAT: simple TCGA data access for integrated statistical analysis in R," *Bioinformatics*, vol. 32, no. 6, pp. 952–954, 2015.

[149] B. P. Durbin, J. S. Hardin, D. M. Hawkins, and D. M. Rocke, "A variance-stabilizing transformation for gene-expression microarray data," *Bioinformatics*, vol. 18, no. suppl_1, pp. S105–S110, 2002.

[150] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, vol. 15, no. 12, p. 550, 2014.

[151] L. Dalton and E. Dougherty, "Application of the bayesian MMSE estimator for classification error to gene expression microarray data," *Bioinformatics*, vol. 27, no. 13, pp. 1822–1831, 2011.

[152] J. P. Holdren *et al.*, "Materials genome initiative for global competitiveness," *National Science and Technology Council OSTP. Washington, USA*, 2011.

[153] N. R. Council *et al.*, *Integrated computational materials engineering: a transformational discipline for improved competitiveness and national security*. National Academies Press, 2008.

[154] D. V. Lindley, *Bayesian Statistics, A Review*. SIAM, 1972.

[155] P. I. Frazier, W. B. Powell, and S. Dayanik, "A knowledge-gradient policy for sequential information collection," *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2410–2439, 2008.

[156] P. I. Frazier, W. B. Powell, and S. Dayanik, "The knowledge-gradient policy for correlated normal beliefs," *INFORMS Journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[157] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[158] B.-J. Yoon, X. Qian, and E. R. Dougherty, "Quantifying the objective cost of uncertainty in complex dynamical systems," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2256–2266, 2013.

[159] R. Dehghannasiri, B.-J. Yoon, and E. R. Dougherty, "Optimal experimental design for gene regulatory networks in the presence of uncertainty," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, pp. 938–950, July 2015.

[160] M. Stone, "Application of a measure of information to the design and comparison of regression experiments," *Annals of Mathematical Statistics*, vol. 30, pp. 55–70, 1959.

[161] M. H. DeGroot, "Uncertainty, information and sequential experiments," *Annals of Mathematical Statistics*, vol. 33, no. 2, pp. 404–419, 1962.

[162] M. H. DeGroot, *Concepts of Information Based on Utility*, pp. 265–275. Dordrecht: Springer Netherlands, 1986.

[163] J. M. Bernardo, "Expected information as expected utility," *Annals of Statistics*, vol. 7, no. 3, pp. 686–690, 1979.

[164] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Stat. Sci.*, vol. 10, no. 3, pp. 273–304, 1995.

[165] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[166] A. Talapatra*, S. Boluki*, T. Duong, X. Qian, E. Dougherty, and R. Arróyave, "Autonomous efficient experiment design for materials discovery with Bayesian model averaging," *Physical Review Materials*, vol. 2, no. 11, p. 113803, 2018.

[167] S. Chen, K.-R. G. Reyes, M. K. Gupta, M. C. McAlpine, and W. B. Powell, "Optimal learning in experimental design using the knowledge gradient policy with application to characterizing nanoemulsion stability," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3, no. 1, pp. 320–345, 2015.

[168] P. I. Frazier and J. Wang, "Bayesian optimization for materials design," *Information science for materials discovery and design*, vol. 225, pp. 45–57, 2016.

[169] Y. Wang, K. G. Reyes, K. A. Brown, C. A. Mirkin, and W. B. Powell, "Nested-batch-mode learning and stochastic optimization with an application to sequential multistage testing in materials science," *SIAM J. Scientific Computing*, vol. 37, 2015.

[170] A. Talapatra, S. Boluki, P. Honarmandi, A. Solomou, G. Zhao, S. F. Ghoreishi, A. Molkeri, D. Allaire, A. Srivastava, X. Qian, *et al.*, "Experiment design frameworks for accelerated discovery of targeted materials across scales," *Frontiers in Materials*, vol. 6, p. 82, 2019.

[171] C. Sima and E. R. Dougherty, "What should be expected from feature selection in small-sample settings," *Bioinformatics*, vol. 22, no. 19, p. 2430, 2006.

[172] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[173] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, no. 3, pp. 409–424, 2009.

[174] D. Madigan, A. E. Raftery, C. Volinsky, and J. Hoeting, "Bayesian model averaging," in *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models, Portland, OR*, pp. 77–83, 1996.

[175] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, pp. 382–401, 1999.

[176] L. Wasserman, "Bayesian model selection and model averaging," *Journal of mathematical psychology*, vol. 44, no. 1, pp. 92–107, 2000.

[177] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[178] M. T. Emmerich, A. H. Deutz, and J. W. Klinkenberg, "Hypervolume-based expected improvement: Monotonicity properties and exact computation," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2147–2154, IEEE, 2011.

[179] S. Aryal, R. Sakidja, M. W. Barsoum, and W.-Y. Ching, "A genomic approach to the stability, elastic, and electronic properties of the MAX phases," *Physica Status Solidi (B)*, vol. 251, no. 8, pp. 1480–1497, 2014.

[180] P. V. Balachandran, D. Xue, J. Theiler, J. Hogden, and T. Lookman, "Adaptive strategies for materials design using uncertainties," *Scientific reports*, vol. 6, 2016.

[181] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[182] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[183] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[184] D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.

[185] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[186] Y. Zhu and N. Zabaras, "Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification," *Journal of Computational Physics*, vol. 366, pp. 415–447, 2018.

[187] L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data," *Journal of Computational Physics*, vol. 425, p. 109913, 2021.

[188] N. Geneva and N. Zabaras, "Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks," *Journal of Computational Physics*, vol. 403, p. 109056, 2020.

[189] S. Karumuri, R. Tripathy, I. Bilionis, and J. Panchal, "Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks," *Journal of Computational Physics*, vol. 404, p. 109120, 2020.

[190] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, "Projection-based model reduction: Formulations for physics-based machine learning," *Computers & Fluids*, vol. 179, pp. 704–717, 2019.

[191] K. Nomura and S. Elghobashi, "The structure of inhomogeneous turbulence in variable density nonpremixed flames," *Theoretical and Computational Fluid Dynamics*, vol. 5, no. 4-5, pp. 153–175, 1993.

[192] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.

[193] L. Sieovich, "Turbulence and the dynamics of coherent structures. part 1: Coherent structures," *Quart Appl Math*, vol. 45, pp. 561–571, 1987.

[194] A. Chatterjee, "An introduction to the proper orthogonal decomposition," *Current science*, pp. 808–817, 2000.

[195] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[196] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*, pp. 1278–1286, 2014.

[197] S. Boluki, R. Ardywibowo, S. Z. Dadaneh, M. Zhou, and X. Qian, "Learnable bernoulli dropout for Bayesian deep learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 3905–3916, PMLR, 2020.

[198] S. Z. Dadaneh and X. Qian, "Bayesian module identification from multiple noisy networks," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2016, no. 1, p. 5, 2016.

[199] M. M. J. Opgenoord, M. Drela, and K. E. Willcox, "Physics-based low-order model for transonic flutter prediction," *AIAA Journal*, vol. 56, no. 4, pp. 1519–1531, 2018.

[200] N. L. Johnson, A. W. Kemp, and S. Kotz, *Univariate discrete distributions*, vol. 444. John Wiley & Sons, 2005.

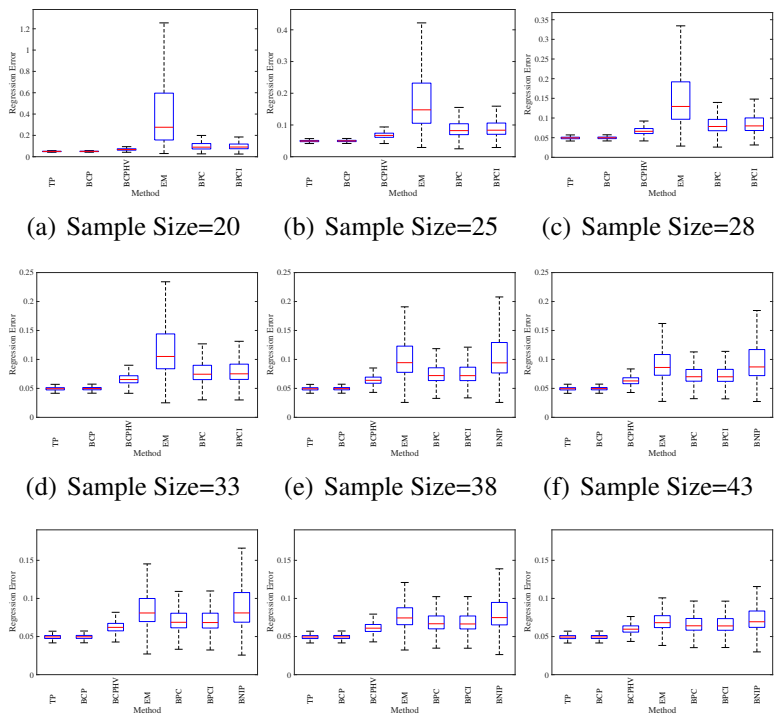
[201] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, 2014.

APPENDIX A

ADDITIONAL RESULTS FOR CHAPTER 3

A.1 More Plots for the Results in Section 3.3.1.3

Box plots of the regression and classification errors over all the networks and all the repetitions for mixing probabilities of 0.72 and 0.28 in Section 3.3.1.3 are shown for different sample sizes in Fig. A.1 and Fig. A.2, respectively.

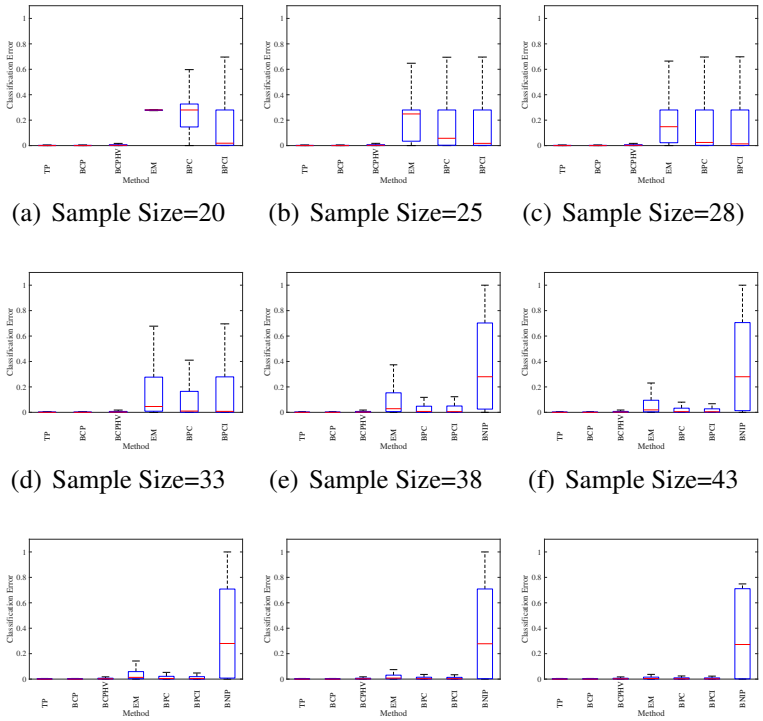


(a) Sample Size=20 (b) Sample Size=25 (c) Sample Size=28

(d) Sample Size=33 (e) Sample Size=38 (f) Sample Size=43

(g) Sample Size=48 (h) Sample Size=55 (i) Sample Size=65

Figure A.1: Box plots of regression errors on synthetic pathways for different sample sizes with $p_1 = 0.72$ and $p_2 = 0.28$.



(a) Sample Size=20 (b) Sample Size=25 (c) Sample Size=28)

(d) Sample Size=33 (e) Sample Size=38 (f) Sample Size=43

(g) Sample Size=48 (h) Sample Size=55 (i) Sample Size=65

Figure A.2: Box plots of classification errors on synthetic pathways for different sample sizes with $p_1 = 0.72$ and $p_2 = 0.28$.

A.2 Single Component Regression Comparison Results

The setup of synthetic pathway and data generation is similar to the procedure described in Section 3.3, but here in each Monte Carlo simulation, the training and test data are generated from only one component. Here, 200 random pathways are simulated, and 40 Monte Carlo repetitions of training and test data generation are done for each fixed pathway and sample size, and in each repetition the average regression error (mean-square error) on 1000 test data points are calculated. For comparison purposes, we have implemented GRACE [82], where in our implementation the regularization parameters are selected based on 10-fold cross validation (as suggested in their paper) in each repetition for each fixed pathway and sample size. The average regression error over all the networks and repetitions as a function of sample size is shown in Fig. A.3. Since only one component exists and there are no missing labels in the data, only one iteration of prior construction is required in our method. Thus, only BPC is compared to GRACE. As can be seen from the figure, BPC outperforms GRACE for small sample sizes and shows a great advantage for very small sample sizes that shrinks as the sample size increases. One thing to note is that GRACE only uses the connectivity information in the pathways, and does not incorporate the regulating information. Also, their method does not consider side information about the relationship of the output and the inputs. In other words, it only considers the information about the connectivity of the predictors in the corresponding pathways (networks). To have a fair comparison, the edges of the pathway that are connected to the regression output were hidden from our method, but still our method can incorporate the regulating relationships in the remaining edges.

A.3 More Plots for the Results in Section 3.3.2

Box plots of regression and classification errors over all repetitions for different sample sizes are shown for noise variance $\sigma_n^2 = 0.05$ in Fig. A.4 and Fig. A.5, respectively. Similar box plots of regression and classification errors for the higher noise variance $\sigma_n^2 = 0.1$ are provided in Fig. A.6 and Fig. A.7, respectively. The average component-conditional classification errors over all the repetitions for both of the components are depicted vs. the sample size for $\sigma_n^2 = 0.05$ in Fig.

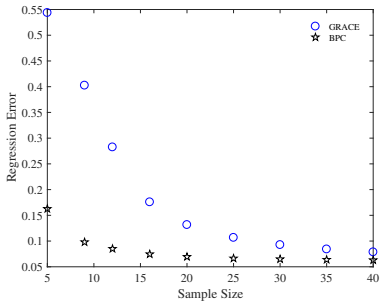
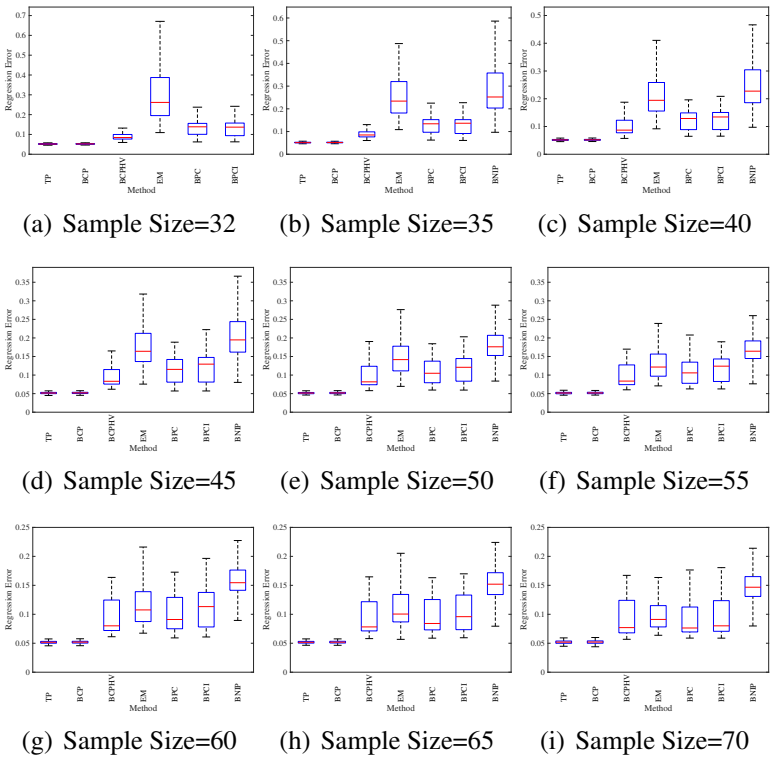


Figure A.3: Average regression error vs sample size in a single component problem.

A.8(a) and Fig. A.8(b), and for $\sigma_n^2 = 0.1$ in Fig. A.9(a) and Fig. A.9(b). Furthermore, the average F-score (geometric mean of precision and recall) over all the repetitions are shown as function of sample size for $\sigma_n^2 = 0.05$ and $\sigma_n^2 = 0.05$ in Fig. A.10 and Fig. A.11, respectively.

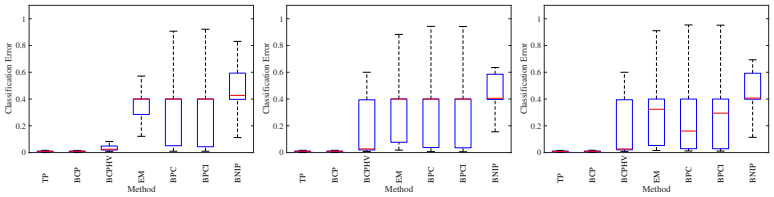


(a) Sample Size=32 (b) Sample Size=35 (c) Sample Size=40

(d) Sample Size=45 (e) Sample Size=50 (f) Sample Size=55

(g) Sample Size=60 (h) Sample Size=65 (i) Sample Size=70

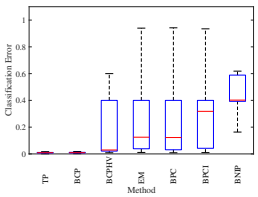
Figure A.4: Box plots of regression errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.05$.



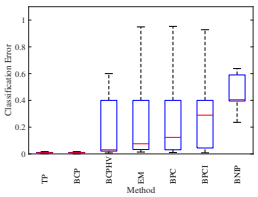
(a) Sample Size=35

(b) Sample Size=40

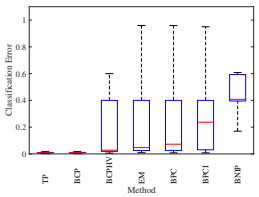
(c) Sample Size=45



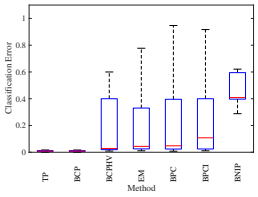
(d) Sample Size=50



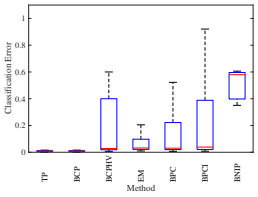
(e) Sample Size=55



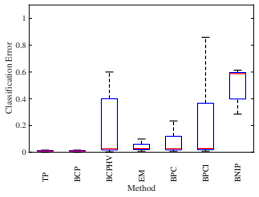
(f) Sample Size=60



(g) Sample Size=65

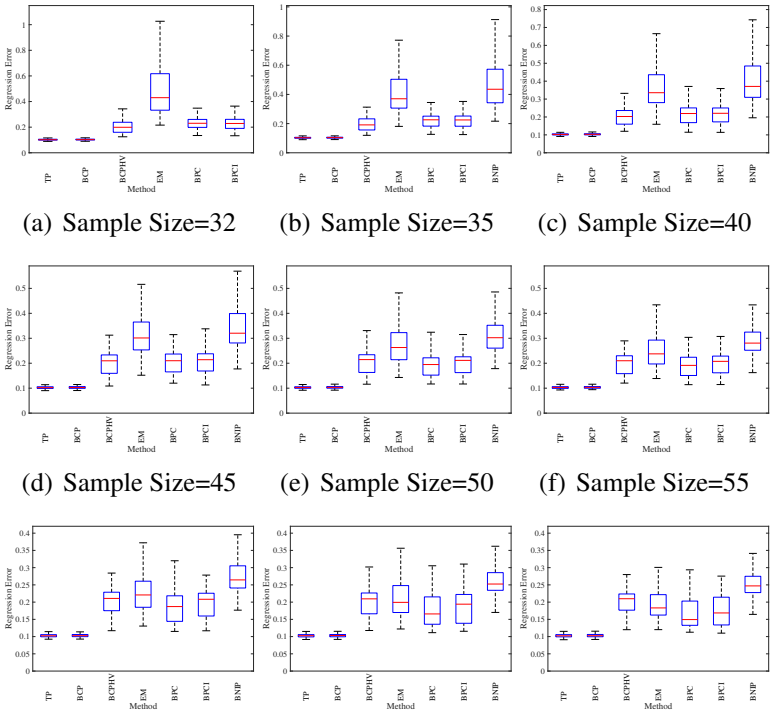


(h) Sample Size=70



(i) Sample Size=75

Figure A.5: Box plots of classification errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.05$.

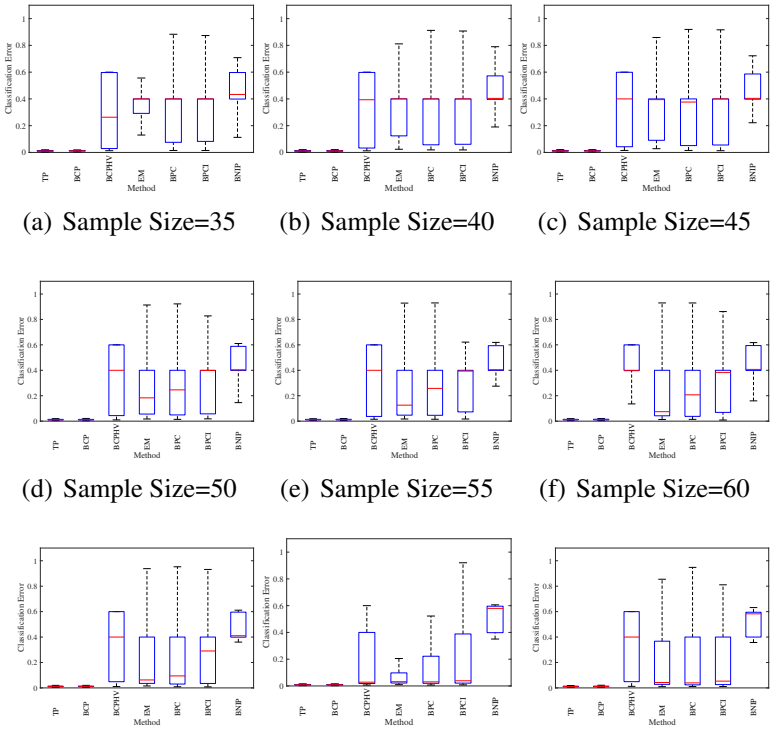


(a) Sample Size=32 (b) Sample Size=35 (c) Sample Size=40

(d) Sample Size=45 (e) Sample Size=50 (f) Sample Size=55

(g) Sample Size=60 (h) Sample Size=65 (i) Sample Size=70

Figure A.6: Box plots of regression errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.1$.

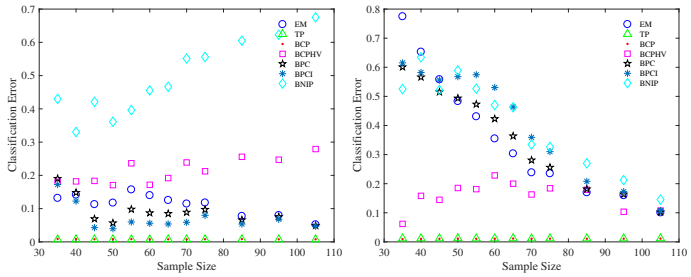


(a) Sample Size=35 (b) Sample Size=40 (c) Sample Size=45

(d) Sample Size=50 (e) Sample Size=55 (f) Sample Size=60

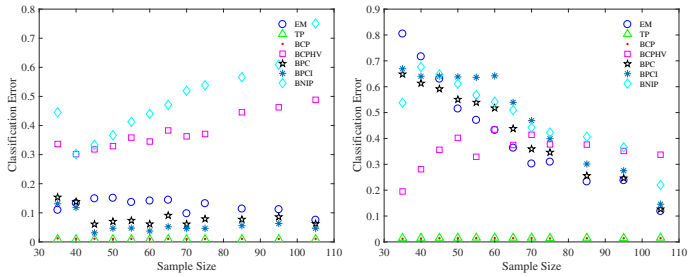
(g) Sample Size=65 (h) Sample Size=70 (i) Sample Size=75

Figure A.7: Box plots of classification errors on colon cancer pathways for different sample sizes with $\sigma_n^2 = 0.1$.



(a) Component 1 (colon. path.) (b) Component 2 (colon. path.)

Figure A.8: Average component-conditional classification errors on colon cancer pathways with $\sigma_n^2 = 0.05$ for the first and second components in the top and bottom panels respectively.



(a) Component 1 (colon. path.) (b) Component 2 (colon. path.)

Figure A.9: Average component-conditional classification errors on colon cancer pathways with $\sigma_n^2 = 0.1$ for the first and second components in the top and bottom panels respectively.

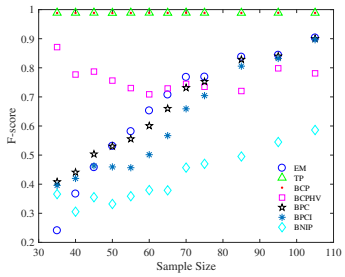


Figure A.10: Average F-score on colon cancer pathways with $\sigma_n^2 = 0.05$.

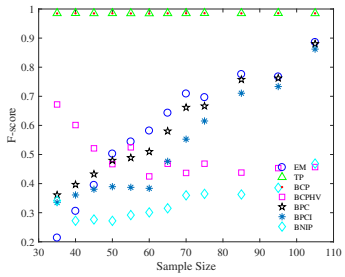


Figure A.11: Average F-score on colon cancer pathways with $\sigma_n^2 = 0.1$.

APPENDIX B

SUPPLEMENTARY MATERIALS FOR CHAPTER 4

B.1 OBSDA Inference via Gibbs Sampling

Sampling $\boldsymbol{\phi}_j$ and $\boldsymbol{\theta}^l_{d,k}$: As shown in [110], the negative binomial random variable $\mathbf{x} \sim \text{NB}(r, p)$ can be generated from a compound Poisson distribution

$$\mathbf{x} = \sum_{t=1}^{T} u_t, \quad u_t \sim \text{Log}(p), \quad T \sim \text{Pois}(-r\ln(1-p)), \tag{B.1}$$

where $u \sim \text{Log}(p)$ being the logarithmic random variable with the probability mass function (PMF) $f_U(u) = -\frac{p^u}{u\ln(1-p)}, u \in \{1, 2, \cdots\}$ [200]. Given $\mathbf{x}$ and r, [110] have shown that an augmented random variable n has a Chinese Restaurant Table (CRT) distribution $\text{CRT}(\mathbf{x}, r)$, which can be generated as $n = \sum_{t=1}^{\mathbf{x}} b_t$, where $b_t \sim \text{Bernoulli}(\frac{r}{r+t-1})$. Utilizing this data augmentation, we can introduce the latent counts as

$$\mathbf{n}^l_{d,j,i}|- \sim \text{CRT}(\mathbf{x}^l_{d,j,i}, \sum_{k=1}^{K} \boldsymbol{\phi}_{j,k}\boldsymbol{\theta}^l_{d,k}). \tag{B.2}$$

From Theorem 4 in [115], the latent counts can be split to subcounts based on a multinomial distribution:

$$(\cdots, \mathbf{n}^l_{d,j,i,k}, \cdots|-) \sim \text{Mult}(\mathbf{n}^l_{d,j,i}; \cdots, \frac{\boldsymbol{\phi}_{j,k}\boldsymbol{\theta}^l_{d,k}}{\sum_{k'=1}^{K} \boldsymbol{\phi}_{j,k'}\boldsymbol{\theta}^l_{d,k'}}, \cdots). \tag{B.3}$$

Based on the multinomial-Dirichlet conjugacy we can update $\boldsymbol{\phi}_j$ as:

$$(\boldsymbol{\phi}_{1,k}, \cdots, \boldsymbol{\phi}_{J,k}|-) \sim \text{Dir}(\eta + \mathbf{n}^{\cdot}_{\cdot,1,\cdot,k}, \cdots, \eta + \mathbf{n}^{\cdot}_{\cdot,J,\cdot,k}), \tag{B.4}$$

where $\mathbf{n}^{\cdot}_{\cdot,j,\cdot,k} = \sum_{d=1}^{D} \sum_{l \in L_d} \sum_{i=1}^{N^l_d} \mathbf{n}^l_{d,j,i,k}$, with L_d, D, and N^l_d denoting the set of labels in domain d, the number of domains, and the number samples in domain d with label l, respectively.

From Proposition 3 in [115], we can generate the latent counts as $\mathbf{n}^l_{d,j,i,k} \sim \text{Pois}(\tilde{p}^l_{d,i}\boldsymbol{\phi}_{j,k}\boldsymbol{\theta}^l_{d,k})$, where $\tilde{p}^l_{d,i} := -\ln(1-p^l_{d,i})$. By the Gamma-Poisson conjugacy we can then update $\boldsymbol{\theta}^l_{d,k}$ as:

$$\boldsymbol{\theta}^l_{d,k}|- \sim \text{Gamma}(u_{d,k} + \mathbf{n}^l_{d,\cdot,\cdot,k}, \frac{1}{v^l + \sum_{i=1}^{N^l_d} \tilde{p}^l_{d,i}}), \tag{B.5}$$

where $\mathbf{n}^l_{d,\cdot,\cdot,k} = \sum_{j=1}^{J}\sum_{i=1}^{N^l_d} \mathbf{n}^l_{d,j,i,k}$.

In our implementation, we approximate $\text{CRT}(\mathbf{x}, r)$ as

$$\begin{aligned}\text{CRT}(\mathbf{x}, r) &= \sum_{t=1}^{\mathbf{m}} \text{Bernoulli}(\frac{r}{r+t-1}) + \sum_{t=m+1}^{\mathbf{x}} \text{Bernoulli}(\frac{r}{r+t-1}) \\ &\approx \text{CRT}(\mathbf{m}, r) + \text{Pois}(r[\Psi(\mathbf{x}+r) - \Psi(\mathbf{m}+r)]),\end{aligned} \tag{B.6}$$

which reduces the computational cost of generating $\mathbf{n}^l_{d,j,i}$ for genes with large count observations. Here, Ψ represents the digamma function.

Sampling $u_{d,k}$: From the property of the Poisson distribution, we have $\mathbf{n}^l_{d,\cdot,i,k} \sim \text{Pois}(\tilde{p}^l_{d,i}\boldsymbol{\theta}^l_{d,k})$. By marginalizing out $\boldsymbol{\theta}^l_{d,k}$ and the CRT data augmentation technique we can write:

$$\mathbf{n}^l_{d,\cdot,i,k} \sim \text{NB}(u_{d,k}, \frac{\tilde{p}^l_{d,i}}{\tilde{p}^l_{d,i} + v^l}), \quad \tilde{\mathbf{n}}^l_{d,i,k} \sim \text{CRT}(\mathbf{n}^l_{d,\cdot,i,k}, u_{d,k}). \tag{B.7}$$

Denoting $\tilde{\tilde{p}}^l_{d,i} = \frac{\tilde{p}^l_{d,i}}{\tilde{p}^l_{d,i}+v^l}$, we have $\tilde{\mathbf{n}}^l_{d,i,k} \sim \text{Pois}(-\ln(1-\tilde{\tilde{p}}^l_{d,i})u_{d,k})$. Thus, we can update $u_{d,k}$ as

$$u_{d,k}|- \sim \text{Gamma}(b_k + \sum_{l\in L_d}\sum_{i=1}^{N^l_d} \tilde{\mathbf{n}}^l_{d,i,k}, \frac{1}{q_d - \sum_{l\in L_d}\sum_{i=1}^{N^l_d} \ln(1-\tilde{\tilde{p}}^l_{d,i})}). \tag{B.8}$$

Sampling b_k: From the property of the Poisson distribution, and by marginalizing out $u_{d,k}$ and the CRT data augmentation technique we can update b_k as

$$b_k|- \sim \text{Gamma}(\frac{\gamma_0}{K} + \sum_{d=1}^{d}\sum_{l\in L_d} \tilde{\tilde{\mathbf{n}}}^l_{d,k}, \frac{1}{c_0 - \sum_{d=1}^{d}\sum_{l\in L_d} \ln(1-\hat{p}^l_d)}), \tag{B.9}$$

where $\hat{p}_d^l = \frac{-\sum_i \ln(1-\tilde{\hat{p}}_{d,i}^l)}{q_d - \sum_i \ln(1-\tilde{\hat{p}}_{d,i}^l)}$, and $\tilde{\hat{\mathbf{n}}}_{d,k}^l \sim \text{CRT}(\tilde{\mathbf{n}}_{d,\cdot,k}^l, b_k)$.

Sampling γ_0: Following a similar procedure as for the update for b_k, by marginalizing out b_k and the CRT data augmentation technique, we can update γ_0 as

$$\gamma_0 | - \sim \text{Gamma}(\alpha_0 + \sum_{l \in L_d} \sum_{k=1}^{K} \hat{\mathbf{n}}_k^l, \frac{1}{\beta_0 - \sum_{l \in L_d} \ln(1 - \hat{\hat{p}}^l)}), \tag{B.10}$$

where $\hat{\hat{p}}^l = \frac{-\sum_d \ln(1-\hat{p}_d^l)}{c_0 - \sum_d \ln(1-\hat{p}_d^l)}$, and $\hat{\mathbf{n}}_k^l \sim \text{CRT}(\tilde{\hat{\mathbf{n}}}_{\cdot,k}^l, \frac{\gamma_0}{K})$.

Sampling v^l*,* q_d*, and* c_0: From the gamma-gamma conjugacy we have:

$$\begin{aligned}
v^l | - &\sim \text{Gamma}(e_0 + \sum_{d=1}^{d} \sum_{k=1}^{K} u_{d,k} \mathbf{1}_{l \in L_d}, \frac{1}{f_0 + \sum_{d=1}^{d} \sum_{k=1}^{K} \boldsymbol{\theta}_{d,k}^l}), \\
q_d | - &\sim \text{Gamma}(w_0 + \sum_{k=1}^{K} b_k, \frac{1}{u_0 + \sum_{k=1}^{K} u_{d,k}}), \\
c_0 | - &\sim \text{Gamma}(a_0 + \gamma_0, \frac{1}{d_0 + \sum_{k=1}^{K} b_k}).
\end{aligned} \tag{B.11}$$

Sampling $p_{d,i}^l$: From the beta-NB conjugacy for the NB probability parameter, we can sample

$$p_{d,i}^l | - \sim \text{Beta}(g_0 + \sum_{j=1}^{J} \mathbf{x}_{d,j,i}^l, h_0 + \sum_{k=1}^{K} \boldsymbol{\theta}_{d,k}^l). \tag{B.12}$$

B.2 Joint Log-Likelihood of SI-OBSDA

$$
\begin{aligned}
\log p(\mathbf{x}, \boldsymbol{z}) = \sum_{d,l,j,i} \Big[& \log\Gamma(\mathbf{x}^l_{d,j,i} + \boldsymbol{\phi}_j^T \boldsymbol{\theta}^l_d) - \log\Gamma(\boldsymbol{\phi}_j^T \boldsymbol{\theta}^l_d) \\
& + \mathbf{x}^l_{d,j,i} \log(p^l_{d,i}) + \boldsymbol{\phi}_j^T \boldsymbol{\theta}^l_d \log(1 - p^l_{d,i}) \Big] \\
+ \sum_{d,l,i} & \Big[(g_0 - 1)\log(p^l_{d,i}) + (h_0 - 1)\log(1 - p^l_{d,i}) \Big] \\
+ \sum_{d,l,k} & \Big[u_{d,k}(\log(\theta^l_{d,k}) + \log(v^l)) - \log(\theta^l_{d,k}) - v^l \theta^l_{d,k} \Big] \\
+ \sum_{d,k} & \Big[b_k(\log(u_{d,k}) + \log(q_d)) - \log(u_{d,k}) - |L_d| \log\Gamma(u_{d,k}) \Big] \\
+ \sum_{j,k} & \Big[- \log(\phi_{j,k}) - \log(1 - \phi_{j,k}) - \log(\sigma_{\phi_{j,k}}) \\
& - \frac{(\log(\phi_{j,k}) - \log(1 - \phi_{j,k}) - \mu_{\phi_{j,k}})^2}{2\sigma^2_{\phi_{j,k}}} \Big] \\
+ \sum_{k} & \Big[(\frac{\gamma_0}{K} - 1)\log(b_k) - c_0 b_k - \log\Gamma(b_k) \Big] \\
+ \sum_{l} & \Big[(e_0 - 1)\log(v^l) - f_0 v^l \Big] + \sum_{d} \Big[(w_0 - 1)\log(q_d) - u_0 q_d \Big] \\
+ \gamma_0 & \log(c_0) - K\log\Gamma(\frac{\gamma_0}{K}) + (\alpha_0 - 1)\log(\gamma_0) - \beta_0 \gamma_0 \\
+ (a_0 & - 1)\log(c_0) - d_0 c_0.
\end{aligned}
\tag{B.13}
$$

B.3 Implementation Remarks for SI-OBSDA

We have implemented SI-OBSDA in TensorFlow, where both $\nabla_{\boldsymbol{\xi}} \underline{\mathcal{L}}_{\tilde{M}}$ and $\nabla_{\boldsymbol{\omega}} \underline{\mathcal{L}}_{\tilde{M}}$ are numerically calculated. Specifically, we update $\boldsymbol{\omega}$ (the parameters of the neural network) and $\boldsymbol{\xi}$ (variational parameters) by the Adam optimizer [201] and gradient descent respectively. In all the experiments we take $\boldsymbol{\epsilon}$ to have the same cardinality of $\boldsymbol{\psi}$, and $T_{\boldsymbol{\omega}}(\boldsymbol{\epsilon})$ as a neural network with three hidden layers with 240, 300, and 240 neurons, and ReLU activation functions. We assume $\tilde{M}$ to be fixed over all epochs and set $\tilde{M} = 50$ in our experiments.

B.4 A Note on the Difference Between the Proposed Model and Variational Autoencoders

Variational autoencoders (VAEs) [195, 196] are widely used for unsupervised feature learning and amortized inference. The canonical encoder in VAEs forces the latent variables to follow a Gaussian distribution which can be restrictive. Also, for VAEs, there is no specific structure to learn useful knowledge for the target domain from source domains and the label information. Additionally, vanilla VAEs do not have the capability of leveraging the prior interactome knowledge.

On the other hand, our proposed model explicitly integrates data from different domains along their labels to learn useful knowledge for the target domain. More specifically, our model utilizes data and label information from multiple domains to learn shared genes embedding as well as domain- and label-dependent latent parameters. The proposed model is specifically designed for (over-dispersed) NGS count data and can take advantage of additional prior knowledge in terms of interaction networks. Moreover, our model learns the latent variables and predictor in a unified fashion, as opposed to a sequential two-step unsupervised feature learning and predictor learning, to be able to learn useful information for the task in the target domain.

It is worth emphasizing that our model is an explicit Bayesian model, and in SI-OBSDA, we only employ neural networks to form a more expressive and flexible variational posterior, where the conditional variational posterior is still explicit, but is mixed with an implicit distribution that uses neural networks.

B.5 Results on Subtyping of Endometrial Carcinoma

In this section, we provide additional results on evaluating the proposed methods for subtyping endometrial carcinoma using RNA-Seq datasets from TCGA. Endometrial carcinoma (UCEC) is one of the most common cancers of the female reproductive system according to the American Cancer Society. The TCGA UCEC dataset contains samples with labels `endometrioid` and `serous endometrial`. It is known that endometrial cancer shares genomic features with serous ovarian cancer [86]. Hence, we examine the target endometrial cancer subtyping accuracy using additional RNA-Seq data from TCGA's study of ovarian serous adenocarcinoma (OV).

For SI-OBSDA we use the same gene-gene network explained in Section 4.3.1 of the main text. Similar as the experiments of the lung cancer subtyping problem in Section 4.3 of the main text, we randomly select the training data, and then we filter out genes with very low total read counts across the training data to only keep the genes with sufficient coverage and read counts that can potentially have useful information for the subtyping problem. Here, we set the filtering threshold to be 10. Finally, we perform differential expression analysis with DESeq2 [121] on the training data, using the default parameters and only the condition covariate, and select 750 out of the top 1500 genes with the highest log-fold change (with gaps of 2) in each experimental run for all the methods for fair comparison.

The target domain contains 221 and 41 samples from endometrioid and serous endometrial, respectively. In each run, we randomly pick 20 samples in total from the target domain for training, 13 with endometrioid endometrial labels and 7 with serous endometrial labels. We use the remaining 34 samples from serous endometrial and a randomly selected subset of size 63 from the remaining endometrioid samples as the test data. We use two different numbers of source samples, 112 and 11, randomly picked from the 299 OV samples in each run for OBSDA, SI-OBSDA, and BMDL. The other baselines including regularized logistic regression (RegLog), regularized linear SVM (Reg SVM), and kernel SVM (SVM) can only use the target training data.

Table B.1: Average errors (in % $\pm$ standard deviations) and AUC ($\pm$ standard deviations) in identifying endometrioid endometrial vs serous endometrial with the source domain containing samples from ovarian serous adenocarcinoma.

Method	$N_s = 11$		$N_s = 112$	
	Error	AUC	Error	AUC
SI-OBSDA	9.27 ± 3.13	93.86 ± 1.72	8.64 ± 2.40	94.91 ± 1.41
OBSDA	10.72 ± 4.05	94.27 ± 2.23	9.07 ± 3.88	95.54 ± 1.71
BMDL	12.09 ± 3.14	94.11 ± 3.03	14.71 ± 4.89	93.05 ± 4.70
Reg Log (t)	23.75 ± 4.52	84.99 ± 7.10	23.75 ± 4.52	84.99 ± 7.10
Reg SVM (t)	16.49 ± 3.36	90.02 ± 2.31	16.49 ± 3.36	90.02 ± 2.31
SVM (t)	17.59 ± 2.37	85.67 ± 4.87	17.59 ± 2.37	85.67 ± 4.87

The error and area under the ROC curve (AUC) for the different methods are shown in Table

B.1. We can see that SI-OBSDA and OBSDA outperform the baselines in terms of error and AUC metrics. Also, they both benefit from more samples from the source domain in training. Overall, the results demonstrate that all methods that can leverage data from the source domain in addition to the target training data perform better than the other baselines that only use the target domain data. Their performance differences are also more prominent in terms of the subtyping error, which more directly relates to our target objective, compared with AUC.

As evidenced by the baselines' performances, this problem seems to be easier than the lung cancer subtyping that is considered in the main text. Nevertheless, the proposed methods still show better performance by using data from the source domain and the prior network knowledge (for SI-OBSDA only) compared with the baselines.