

Non-parametric Bayesian Learning with Incomplete Data

by

Chunping Wang

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Lawrence Carin, Supervisor

Jeffrey Krolak

Rebecca Willett

David Dunson

Mauro Maggioni

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2010

ABSTRACT
(Electrical Engineering)

Non-parametric Bayesian Learning with Incomplete Data

by

Chunping Wang

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Lawrence Carin, Supervisor

Jeffrey Krolik

Rebecca Willett

David Dunson

Mauro Maggioni

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University
2010

Copyright © 2010 by Chunping Wang
All rights reserved

Abstract

In most machine learning approaches, it is usually assumed that data are complete. When data are partially missing due to various reasons, for example, the failure of a subset of sensors, image corruption or inadequate medical measurements, many learning methods designed for complete data cannot be directly applied. In this dissertation we treat two kinds of problems with incomplete data using non-parametric Bayesian approaches: classification with incomplete features and analysis of low-rank matrices with missing entries.

Incomplete data in classification problems are handled by assuming input features to be generated from a mixture-of-experts model, with each individual expert (classifier) defined by a local Gaussian in feature space. With a linear classifier associated with each Gaussian component, nonlinear classification boundaries are achievable without the introduction of kernels. Within the proposed model, the number of components is theoretically “infinite” as defined by a Dirichlet process construction, with the actual number of mixture components (experts) needed inferred based upon the data under test. With a higher-level DP we further extend the classifier for analysis of multiple related tasks (multi-task learning), where model components may be shared across tasks. Available data could be augmented by this way of information transfer even when tasks are only similar in some local regions of feature space, which is particularly critical for cases with scarce incomplete training samples from each task. The proposed algorithms are implemented using efficient

variational Bayesian inference and robust performance is demonstrated on synthetic data, benchmark data sets, and real data with natural missing values.

Another scenario of interest is to complete a data matrix with entries missing. The recovery of missing matrix entries is not possible without additional assumptions on the matrix under test, and here we employ the common assumption that the matrix is low-rank. Unlike methods with a preset fixed rank, we propose a non-parametric Bayesian alternative based on the singular value decomposition (SVD), where missing entries are handled naturally, and the number of underlying factors is imposed to be small and inferred in the light of observed entries. Although we assume missing at random, the proposed model is generalized to incorporate auxiliary information including missingness features. We also make an attempt to acquire new entries actively based on the uncertainty manifested by posteriors. By introducing a probit link function, we are able to handle counting matrices with the decomposed low-rank matrices latent. The basic model and its extensions are validated on synthetic data, a movie-rating benchmark and a new data set presented for the first time.

Contents

| | |
|--|-------------|
| Abstract | iv |
| List of Tables | ix |
| List of Figures | x |
| List of Abbreviations and Symbols | xv |
| Acknowledgements | xvii |
| 1 Introduction | 1 |
| 1.1 Machine Learning Basics | 1 |
| 1.2 Bayesian Modeling | 3 |
| 1.3 Mixture of Experts | 4 |
| 1.4 Supervised Learning with Incomplete Data | 6 |
| 1.5 Multi-task Learning | 10 |
| 1.6 Collaborative Filtering | 13 |
| 1.7 Summary of the Remaining Chapters | 15 |
| 2 Dirichlet Process and Its Extensions | 17 |
| 2.1 Dirichlet Process | 17 |
| 2.1.1 Pólya urn scheme | 18 |
| 2.1.2 Stick-breaking construction | 20 |
| 2.2 Dirichlet Process Mixtures | 22 |
| 2.3 Hierarchical Dirichlet Process | 23 |

| | | |
|----------|--|-----------|
| 3 | Classification with Incomplete Data | 24 |
| 3.1 | Introduction | 24 |
| 3.2 | Infinite Quadratically Gated Mixture of Experts | 26 |
| 3.2.1 | Quadratically gated mixture of experts | 26 |
| 3.2.2 | Infinite QGME via Dirichlet process | 28 |
| 3.2.3 | A variant for high-dimensional problems | 30 |
| 3.3 | Incomplete Data Problem | 32 |
| 3.4 | Variational Bayesian Inference | 35 |
| 3.4.1 | Basic construction | 35 |
| 3.4.2 | Variational distributions specification and updating | 37 |
| 3.4.3 | Prediction | 42 |
| 3.4.4 | Computational complexity | 44 |
| 3.5 | Experimental Results | 45 |
| 3.5.1 | Synthetic data | 46 |
| 3.5.2 | Benchmark data | 49 |
| 3.5.3 | Unexploded ordnance data | 56 |
| 3.5.4 | Sepsis classification data | 58 |
| 3.6 | Summary | 60 |
| 4 | Multi-task Classification with Incomplete Data | 61 |
| 4.1 | Introduction | 61 |
| 4.2 | Multi-Task Learning via the Hierarchical Dirichlet Process | 62 |
| 4.3 | Variational Bayesian Inference | 64 |
| 4.4 | Experimental Results | 69 |
| 4.4.1 | Landmine Detection Data | 70 |
| 4.4.2 | Handwritten letters data | 77 |

| | | |
|----------|---|------------|
| 4.5 | Summary | 79 |
| 5 | Bayesian Matrix Completion | 80 |
| 5.1 | Introduction | 80 |
| 5.2 | Bayesian Singular Value Decomposition | 81 |
| 5.3 | Inference by Markov Chain Monte Carlo | 84 |
| 5.4 | Model Generalizations | 88 |
| 5.4.1 | Auxiliary information | 88 |
| 5.4.2 | Active learning | 89 |
| 5.4.3 | Probit link function | 90 |
| 5.5 | Example Results | 92 |
| 5.5.1 | Parameter settings | 92 |
| 5.5.2 | Synthetic matrices | 93 |
| 5.5.3 | Movie ratings | 97 |
| 5.5.4 | MLB data from 1954-2008 | 103 |
| 5.6 | Summary | 107 |
| 6 | Conclusions and Future Work | 108 |
| 6.1 | Conclusions and Discussions | 108 |
| 6.2 | Future Work | 110 |
| | Bibliography | 111 |
| | Biography | 124 |

List of Tables

| | | |
|-----|---|-----|
| 3.1 | Details of Ionosphere and WDBC data sets. | 49 |
| 4.1 | Handwritten letters classification data set. | 77 |
| 5.1 | Auxiliary features in 1M MovieLens data set. | 98 |
| 5.2 | Prediction errors on 1M MovieLens data set. Mean and standard deviation over the three partitions are reported. For the proposed BSVD model, the codes “0000”, “0011”, “1100” and “1111” correspond to the basic model as in (5.3), the models with user auxiliary information, movie auxiliary information, and all available auxiliary information, respectively. Results for the GP-LVM are cited from [LU09], with the latent dimensionality yielding these best results indicated. | 100 |
| 5.3 | MLB data log-likelihood for the year of 2008, evaluated for pitchers and batters appearing in training sets: 2007, 2003 to 2007, and 1954 to 2007, respectively. | 104 |
| 5.4 | Prediction on probabilities of top batters being successful against top pitchers (means and standard deviations). Top batters: batting in at least the median number of events among all batters in a given year, and with the highest empirical hitting rates; top pitchers: pitching in at least the median number of events among all pitchers in a given year, and with the lowest empirical hitting rates. | 106 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | An example with complex class boundary | 6 |
| 3.1 | Graphical representation of the iQGME for single-task learning. All circles denote random variables, with shaded ones indicating observable data, and bright ones representing hidden variables. Diamonds denote fixed hyper-parameters, boxes represent independent replicates with the numbers of copies indicated at the lower-right corner, and arrows indicate the dependence between variables (pointing from parents to children). | 30 |
| 3.2 | Synthetic three-Gaussian single-task data with inferred components. (a) Data in feature space with true labels and true Gaussian components indicated; (b) inferred posterior expectation of weights on components, with standard deviations depicted as error bars; (c) ground truth with posterior means of dominant components indicated (the linear classifiers and Gaussian ellipses are inferred from the data). | 46 |
| 3.3 | Synthetic three-Gaussian single-task data: (a) prior and posterior beliefs on the number of dominant components; (b) prior and posterior beliefs on α | 47 |
| 3.4 | Synthetic three-Gauss single-task data: (a) prediction in feature space using the full posteriors; (b) prediction in feature space using the posterior means; (c) a common broad prior on local experts; (d) variational posteriors on local experts. | 48 |

| | | |
|-----|--|----|
| 3.5 | Results on Ionosphere data set for (a)(b) 25%, and (c)(d) 50% of the feature values missing. For legibility, we only report the standard deviation for the proposed iQGME-VB algorithm as error bars, and present the compared algorithms in two figures for each case. The results of the finite QGME solved with an expectation-maximization method are cited from [LLC07a], and those of LR-Integration are cited from [WLX ⁺ 07]. Since the performance of the QGME-EM is affected by the choice of number of experts K , the overall best results among $K = 1, 3, 5, 10, 20$ are cited for comparison in each case (no such selection of K is required for the proposed iQGME-VB algorithm). | 51 |
| 3.6 | Results on WDBC data set for cases when (a)(b) 25%, and (c)(d) 50% of the feature values are missing. Refer to Figure 3.5 for additional information. | 52 |
| 3.7 | The comparison on the Ionosphere data set between QGME-EM with different preset number of clusters K and the proposed iQGME-VB, when (a)(b)(c) 25%, and (d)(e)(f) 50% of the features are missing. In each row, 10%, 50%, and 90% of samples are used for training, respectively. Results of QGME-EM are cited from [LLC07a]. | 53 |
| 3.8 | Number of clusters for the Ionosphere data set inferred by iQGME-VB. (a) Prior and inferred posterior on the number of clusters for one trial given 10% samples for training. The number of clusters for the case when (b) 25%, and (c) 50% of features are missing. The most probable value of clusters number is used for each trial to generate (b) and (c) (e.g, the most probable value of clusters number is two for the trial shown in (a)). In (b) and (c), the distribution of number of clusters for the ten trials given each missing fraction and training fraction is presented as a box-plot, where the red line represents the median; the bottom and top of the blue box are the 25th and 75th percentile, respectively; the bottom and top black lines are the end of the whiskers, which could be the minimum and maximum, respectively; if some data are beyond 1.5 times of the length of the blue box (interquartile range), they are outliers, indicated by a red '+'. | 54 |
| 3.9 | Ratio of missing values whose true values are less than one standard deviation (red circles) or two standard deviations (blue squares) away from the posterior means for the Ionosphere data set with 25% feature values missing. One trial for each training size is considered. | 55 |

| | | |
|------|---|----|
| 3.10 | Comparison between VB and MCMC inferred iQGME on the Ionosphere data with 25% features missing in terms of (a) performance, (b) time consumed for each iteration, and (c) number of iterations. For the VB inference, we set a threshold (10^{-6}) for the relative change of lower bound in two consecutive iterations as the convergence criterion; for the MCMC inference, we discard the initial samples from the first 1000 iterations (burn-in), and collect the next 500 samples to present the posterior. | 56 |
| 3.11 | Missing pattern for the unexploded ordnance data set, where black and white indicate observed and missing, respectively. | 57 |
| 3.12 | Mean performance over 100 random training/test partitions for each training fraction on the unexploded ordnance data set, in terms of (a) area under the ROC curve, and (b) classification accuracy. | 58 |
| 3.13 | Sepsis data set. (a) Missing pattern, where black and white indicate observed and missing, respectively, (b) mean performance over 100 random training/test partitions for each training fraction. | 59 |
| 4.1 | Graphical representation of the iQGME for multi-task learning via the hierarchical Dirichlet process (HDP). Refer to Figure 3.1 for additional information. | 65 |
| 4.2 | Number of landmines and clutter in each task for the landmine-detection data set [XLCK07]. | 71 |
| 4.3 | Average AUC over 19 tasks of landmine detection with complete data. Error bars reflect the standard deviation across 100 random partitions of training and test subsets. Results of logistic regression based algorithms are cited from [XLCK07], where LR-MTL and LR-STL respectively correspond to SMTL-2 and STL in Figure 3 of [XLCK07]. | 72 |
| 4.4 | Similarity between tasks in the landmine detection problem with complete data given (a) 20, (b) 100, and (c) 300 training samples from each task. The size of green blocks represent the value of the corresponding matrix element. | 74 |
| 4.5 | Average AUC over 19 tasks of landmine detection for the cases when (a) 25%, (b) 50%, and (c) 75% of the features are missing. Mean values of performance across 10 random partitions of training and test subsets are reported. Error bars are omitted for legibility. | 75 |

| | | |
|-----|--|-----|
| 4.6 | Similarity between tasks in the landmine detection problem with incomplete data. Row 1, 2 and 3 corresponds to the cases with 25%, 50% and 75% features missing, respectively; column 1, 2 and 3 corresponds to the cases with 20, 100 and 300 training samples from each task, respectively. | 76 |
| 4.7 | Sample images of the handwritten letters. The two images in each column represents the two classes in the corresponding task described in Table 4.1. | 77 |
| 4.8 | Average AUC over eight tasks of handwriting letters classification for the cases when (a) none, (b) 25%, (c) 50%, and (d) 75% of the features are missing. Mean values of performance with one standard deviation across 10 random partitions of training and test subsets are reported. | 78 |
| 5.1 | Graphical representation of the BSVD model. Refer to Figure 3.1 for additional information. | 84 |
| 5.2 | Empirical recovery rate of full matrices over 50 random trials for each rank r and number of observed entries m (matrices size $I = 50$). The degrees of freedom $d_r = r(2I - r)$. A matrix \mathbf{Y} is declared to be recovered if the reconstructed matrix $\hat{\mathbf{Y}}$ satisfies $\ \hat{\mathbf{Y}} - \mathbf{Y}\ _F / \ \mathbf{Y}\ _F < 10^{-3}$. Results are for (a) [CR09] and (b) BSVD ($a = b = 1$). The green lines are equal-rank trajectories with the rank increasing monotonously from the lower left to the upper right. | 94 |
| 5.3 | Empirical recovery rate of full matrices given by the BSVD when (a) $a = 1, b = 10$ and (b) $a = 10, b = 1$. Refer to Figure 5.2 for additional information. | 95 |
| 5.4 | Sequentially acquire entries from noise-free matrices ($I = 50$) of rank (a) $r = 1$, (b) $r = 5$, and (c) $r = 10$. Blue corresponds to uniformly sampling entries at random; red represents selecting entries actively. Error bars reflect the standard deviation of reconstructed errors over 10 independent trials. | 96 |
| 5.5 | Sequentially acquire entries from noisy matrices ($I = 50$). (a) $SNR = 20dB, r = 5$, (b) $SNR = 20dB, r = 10$, (c) $SNR = 10dB, r = 5$, and (d) $SNR = 10dB, r = 10$ | 97 |
| 5.6 | 1M MovieLens: (a) RMSE and (b) NMAE as a function of the fraction of data used for training (five random selections of training data for each training size). The results of the GP-LVM are cited from Figure 2 in [LU09]. To make the figures legible, we present only the performance means for the GP-LVM with 2D, 6D and 10D latent space, which are representative. The performance means and standard deviations are reported for the proposed BSVD model. | 102 |

| | | |
|-----|--|-----|
| 5.7 | 1M MovieLens: Inferred rank by the proposed BSVD model as a function of data used for training. Each box plot represents the distribution of the rank inferred in five trials for one training size. Detailed explanation for box plots is in the caption of Figure 3.8. | 103 |
| 5.8 | Average standard deviation of predicted probabilities of batters being successful against pitchers (1954-2008). Batters are sorted in ascending order of the total number of events each has participated in. The total number of events for a given batter ranges from 1 to 16492 with a median 143 (half of the batters have less than 143 at-bats). | 106 |

List of Abbreviations and Symbols

| | |
|--------|--|
| AUC | area under the ROC curve. |
| BPMF | Bayesian probabilistic matrix factorization. |
| BSVD | Bayesian singular value decomposition. |
| CF | collaborative filtering. |
| DP | Dirichlet process. |
| DPM | Dirichlet process mixtures. |
| EM | expectation-maximization. |
| EMI | electromagnetic induction. |
| GMM | Gaussian mixture model. |
| GP | Gaussian process. |
| GP-LVM | Gaussian process linear variable models. |
| HDP | Hierarchical Dirichlet process. |
| HMM | hidden Markov model. |
| i.i.d. | independently and identically distributed. |
| iQGME | infinite Quadratically gated mixture of experts. |
| KL | Kullback-Leibler. |
| KNN | K -nearest-neighbors. |
| KSBP | kernel stick-breaking process. |
| LR | logistic regression. |
| MAR | missing at random. |

| | |
|------|---|
| MCMC | Markov chain Monte Carlo. |
| ME | mixture of experts. |
| MFA | mixtures of factor analyzers. |
| MI | multiple imputation. |
| ML | maximum likelihood. |
| MLB | Major League Baseball. |
| MNAR | missing not at random. |
| MTL | multi-task learning. |
| NMAE | normalized mean absolute error. |
| PCA | principal components analysis. |
| QGME | Quadratically gated mixture of experts. |
| RBF | radial basis function. |
| RMSE | root mean squared error. |
| ROC | receiver operating characteristic. |
| RVM | relevance vector machine. |
| STL | single-task learning. |
| SVD | singular value decomposition. |
| SVM | support vector machine. |
| UXO | unexploded ordnance. |
| VB | variational Bayesian. |

Acknowledgements

First of all I would like to express my deepest gratitude to my advisor, Professor Lawrence Carin, for his invaluable guidance, encouragement, support and endless patience throughout my graduate studies. Without his brilliant guidance, this dissertation could not be completed.

I would like to thank Professor David Dunson, for his illuminating instruction and valuable feedback on my research work. I would like to thank Professor Jeffrey Krolik, Professor Rebecca Willett, and Professor Mauro Maggioni, for kindly taking time to serve on my committee, and to provide thoughtful comments.

My thanks and appreciation also go to all my colleagues in our research group: Dr. Xuejun Liao, Dr. Ya Xue, Dr. David Williams, Dr. Hui Li, Dr. Yuting Qi, Dr. Qiuhua Liu, Dr. Qi An, Dr. Kai Ni, Dr. Dehong Liu, Dr. Iulian Pruteanu, Dr. John Paisley, Dr. Lu Ren, Dr. Lan Du, Dr. Bo Chen, Haojun Chen, Minhua Chen, Minyuan Zhou and Eric Wang. I have benefited a lot from enlightening discussions with them, and I would like to thank them for their friendship, help and encouragement. I enjoy the time spent with them during these years at Duke.

Finally, and most importantly, I would like to thank my parents for their lifelong love and support. I am so proud to have my dear husband Lihan who has been standing by me with his love, support and encouragement and my little son Leo who is a source of unending love and joy. I would also like to thank my parents-in-law for their consideration, help and support. This dissertation is dedicated to them.

Introduction

1.1 Machine Learning Basics

As a field of increasing interest, machine learning is concerned with the development of algorithms and techniques that allow computers to “learn”. A learning process may involve *predictors* \mathbf{x} , *responses* y and the *mapping rule* $f(\mathbf{x})$. There are two broad families of basic machine learning problems, categorized according to whether responses y are present or not.

If no responses (y) are imposed, the problem is referred to as *unsupervised* learning, for which the purpose is to discover the underlying structure of the predictors without any “supervisor”. In this class of problems, the intrinsic statistical characteristics of predictors are usually of interest, and hence generative models such as the hidden Markov model (HMM) [BP66] and the Gaussian mixture model (GMM) are typically used. There are many applications of unsupervised learning, for example, image segmentation [MS01] and music analysis [QPC07].

In contrast to unsupervised learning, in *supervised* learning the mapping rule

from predictors to corresponding response variables is of interest. If the responses are continuous quantities, e.g., customer income, insurance price or patient blood pressure, this is termed a *regression* problem. As a special case when the responses are class types, e.g., dog, cat or squirrel, it is a *classification* problem. Each object (e.g., a customer, a patient, an animal) is a *sample*. In the context of classification, predictors are usually referred to as *features*, and discrete response variables and the mapping rule are termed *labels* and the *classifier*, respectively.

The goal of supervised learning is to infer the labels for unlabeled samples, based on labeled (training) data with a common assumption that the labeled and unlabeled data were generated from the same underlying process. For example, after we see some animals and are told which of them are dogs, cats and squirrels, we learn a rule to distinguish them from each other. If we assume that a new unknown animal is not totally different from the ones we are familiar with, then we may tell which class it belongs to. Here the rule is a classifier, and the first group of animals that is used to learn the rule is called *training data*. The new unknown animal is one instance of *test data*, which could be used to quantify the performance of the classifier. Since test samples are usually not identical with the training samples, the mapping rule must be capable of generalization, rather than simply memorizing the exact features of the training data, which may lead to over-fitting.

Sometimes the functional form of the underlying generative model or the mapping rule to be learned is specified, with some parameters to be determined during the learning procedure. Machine learning methods developed in this manner are referred to as being parametric. Many state-of-art methods fall into this category, such as the support vector machine (SVM) [Vap95], the relevance vector machine (RVM) [Tip00], the hidden Markov model (HMM) [BP66], and the Gaussian mixture model (GMM). By contrast the term ‘non-parametric’ has at least two different meanings. First, those methods without any parametric form could be referred to as ‘non-

parametric’ approaches, such as K -nearest-neighbors (KNN) and K -means. The second (emerging) meaning of non-parametric is that the structure of a model is not assumed to be fixed. In these techniques, a parametric form is usually assumed but the model may grow in size to accommodate the complexity of the data. Typical examples include Gaussian process regression [WR96] and Dirichlet process mixtures [Fer73]. The second kind of non-parametric models are what we consider in this dissertation.

1.2 Bayesian Modeling

Generally speaking, a *model* we build for a learning task could be either a deterministic objective function up to optimization, as in the support vector machine (SVM) [Vap95], or some probabilistic distribution, as in the relevance vector machine (RVM) [Tip00]. For example, in regression problems, one may minimize $\|y - f(\mathbf{x}; \boldsymbol{\theta})\|$ with some regularization on *parameters* $\boldsymbol{\theta}$; alternatively, a stochastic noise term (typically normally distributed, i.e., $\epsilon \sim \mathcal{N}(0, \tau^{-1})$) may be introduced such that

$$y = f(\mathbf{x}; \boldsymbol{\theta}) + \epsilon, \tag{1.1}$$

and accordingly

$$y \sim \mathcal{N}(f(\mathbf{x}; \boldsymbol{\theta}), \tau^{-1}). \tag{1.2}$$

Because of the uncertainty associated with the observed samples and the inevitable noise introduced during measurement, statistical analysis has become an important research direction in recent years.

More specifically, those *parameters* appearing in statistical models (e.g., (1.2)) could be regarded as unknown but fixed numbers; however, from a Bayesian perspective, no underlying “true” values of those parameters exist, and one can only have

some *belief* on the values of the parameters $\boldsymbol{\theta}$. Before seeing any data, one may have some *prior* belief, which could be modified to be *posterior* belief once some data are observed. For a *likelihood* model as in (1.2), a simple Bayesian analysis starts with imposing a prior probability $p(\boldsymbol{\theta})$. A posterior probability $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$ could be computed using Bayes' rule $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y}) = p(\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})/p(\mathbf{y}|\mathbf{x}) \propto p(\boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. While searching for the point-estimated “true” values for parameters, we may take the risk of over-fitting the limited observed data. On the contrary, the uncertainty on observed samples is manifested by the uncertainty on the model parameters (the posterior distribution) in Bayesian framework. As a result, Bayesian models are generally more powerful for generalizing to test samples which are not observed when performing model training. Besides counting the uncertainty on model parameters, Bayesian modeling also allows for uncertainty on model structure and it favors the simplest model which fits the observed data [Mac03].

In many situations, one may consider that the prior on $\boldsymbol{\theta}$ depends on other parameters ϕ . Accordingly, the prior $p(\boldsymbol{\theta})$ is replaced by a prior $p(\boldsymbol{\theta}|\phi)$, and a prior $p(\phi)$ on the newly introduced parameters ϕ may be imposed. A resulting posterior probability $p(\boldsymbol{\theta}, \phi|\mathbf{x}, \mathbf{y}) \propto p(\phi)p(\boldsymbol{\theta}|\phi)p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. This is a simple example of a *hierarchical Bayes model*. The process may be repeated, for example, by further introducing additional parameters ψ , which appear in the prior of ϕ and have their own prior. Eventually the process terminates at some layer with the hyper-parameters preset.

1.3 Mixture of Experts

It may be desirable to assume a simple linear form for the mapping function $f(\mathbf{x}; \boldsymbol{\theta})$ in many applications, i.e., $f(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}$. Therefore, (1.1) reduces to

$$y = \boldsymbol{\theta}^T \mathbf{x} + \epsilon, \tag{1.3}$$

As a special case, in classification the outputs y are discrete and a link function σ (e.g., logistic link or probit link) is usually employed to map the real numbers to probabilities in $[0, 1]$. That is

$$P(y = 1) = \sigma(\boldsymbol{\theta}^T \mathbf{x}). \quad (1.4)$$

The line (or hyperplane in high dimensional feature space) $\boldsymbol{\theta}^T \mathbf{x} = 0$ is often used as the decision boundary in feature space, assuming equal (symmetric) costs associated with the two hypotheses.

The linear model for regression/classification is widely used since it is simple and easy to implement. However, in many classification applications, the desirable class boundary may be more complex (e.g., see Figure 1.1 for a toy example) than a straight line or a hyperplane, which could be provided by a global linear classifier as in (1.4). Although we may introduce some kernel function to produce nonlinear decision boundary in the original feature space, it may take effort to choose an appropriate kernel for a global classifier to fit a complicated decision boundary everywhere. It will be much more efficient if we decompose such a complicated classification task into multiple simple sub-tasks, and tackle each of them locally in feature space.

This “divide and conquer” idea has been attracting increasing attention recently in various fields including statistics and machine learning. As a well known and early example, the classification and regression tree (CART) proposed by Breiman, Friedman, Olshen and Stone [BFOS84] provides hard splits of the feature space and the partitioning is restricted to be parallel to coordinates. The main concern of hard splits is that the variance of the learning algorithm becomes larger because a smaller amount of data are taken into account in each local region. Jacobs, Jordan, Nowlan and Hinton [JJNH91] proposed a statistical model known as the *mixture of experts* (ME), which is essentially an input conditional mixture model. The mixture of experts (ME) [JJNH91] consists of a set of experts, which describe the conditional

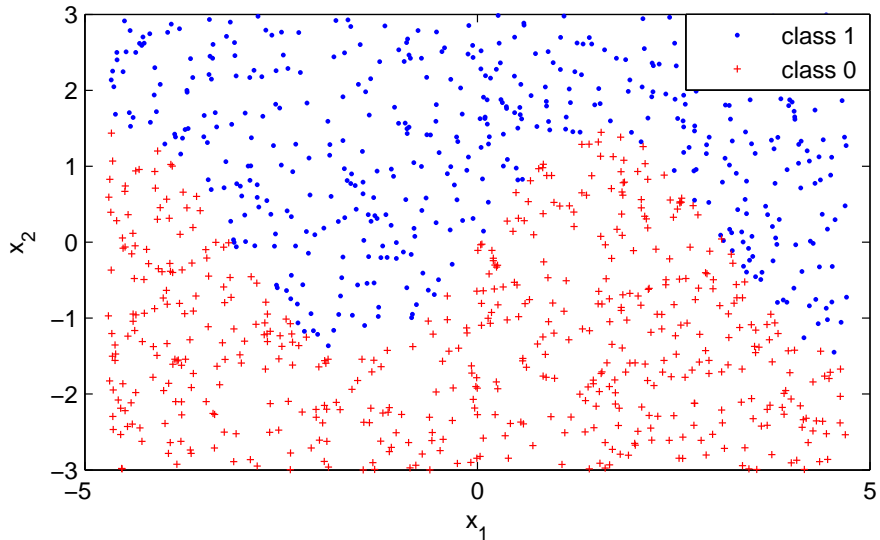


FIGURE 1.1: An example with complex class boundary

distribution of the output in local regions, and a gate which defines the probabilities of choosing different local experts. The ME model provides soft partitions and the feature space can be split by the gating node at any orientation. However, the mixture of experts (ME) [JJNH91] and many of its extensions [WR94, XJH95] have a model-selection issue, since the learning phase has to rely on a preset structure. This issue could be addressed via a Dirichlet process (DP) [Fer73] prior as discussed in detail later.

1.4 Supervised Learning with Incomplete Data

In many applications one must deal with data that have been collected incompletely. For example, in censuses and surveys, some participants may not respond to certain questions [Rub87]; in email spam filtering, server information may be unavailable for emails from external sources [DHS08]; in medical studies, measurements on some subjects may be partially lost at certain stages of the treatment [Ibr90]; in DNA

analysis, gene-expression microarrays may be incomplete due to insufficient resolution, image corruption, or simply dust or scratches on the slide [WLJF06]; in sensing applications, a subset of sensors may be absent or fail to operate at certain regions [WC05]. In these applications, we are interested in prediction of responses for new samples with features (inputs) partially missing. Since most supervised learning procedures (for example, regression and classification) are designed for complete data, and cannot be directly applied to incomplete data, the appropriate handling of missing data is challenging.

Traditionally, data are often “completed” by *ad hoc* editing, such as case deletion and single imputation, where feature vectors with missing values are simply discarded or completed with specific values in the initial stage of analysis, before the main inference (for example, mean imputation and regression imputation [SG02]). Although analysis procedures designed for complete data become applicable after these edits, shortcomings are clear. For case deletion, discarding information is generally inefficient, especially when data are scarce. Secondly, the remaining complete data may be statistically unrepresentative. More importantly, even if the incomplete-data problem is eliminated by ignoring data with missing features in the training phase, it is still inevitable in the test stage since test data cannot be ignored simply because a portion of features are missing. For single imputation, the main concern is that the uncertainty of the missing features is ignored by imputing fixed values.

The work of [Rub76] developed a theoretical framework for incomplete-data problems, where widely-cited terminology for missing patterns was first defined. It was proven that ignoring the *missing mechanism* is appropriate [Rub76] under the *missing at random* (MAR) assumption, meaning that the *missing mechanism* is conditionally independent of the missing features given the observed data. As elaborated later, given the MAR assumption [DHS08, Ibr90, WC05], incomplete data can generally be handled by full maximum likelihood and Bayesian approaches; however,

when the *missing mechanism* does depend on the missing values (*missing not at random* or MNAR), a problem-specific model is necessary to describe the *missing mechanism*, and no general approach exists. In this dissertation we address missing features under the MAR assumption. Previous work in this setting may be placed into two groups, depending on whether the missing data are handled before algorithm learning or within the algorithm.

For the former, an extra step is required to estimate $p(\mathbf{x}^m|\mathbf{x}^o)$, the conditional distributions of missing values given observed ones, with this step distinct from the main inference algorithm. After $p(\mathbf{x}^m|\mathbf{x}^o)$ is learned, various imputation methods may be performed. As a Monte Carlo approach, Bayesian multiple imputation (MI) [Rub87] is widely used, where multiple ($M > 1$) samples from $p(\mathbf{x}^m|\mathbf{x}^o)$ are imputed to form M “complete” data sets, with the complete-data algorithm applied on each, and results of those imputed data sets combined to yield a final result. The MI method “completes” data sets so that algorithms designed for complete data become applicable. Furthermore, [Rub87] showed that MI does not require as many samples as Monte Carlo methods usually do. With a mild Gaussian mixture model (GMM) assumption for the joint distribution of observed and missing data, [WLX⁺07] managed to analytically integrate out missing values over $p(\mathbf{x}^m|\mathbf{x}^o)$ and performed essentially infinite imputations. Since explicit imputations are avoided, this method is more efficient than the MI method, as suggested by empirical results [WLX⁺07]. Other examples of these two-step methods include [WC05, SVH05, SBS06].

Since the imputation models are estimated separately from the main inference algorithm, much flexibility is left for selection of the subsequent algorithm, with few modifications needed for the complete-data approaches. However, as discussed further below, learning two subsets of unknowns separately (density functions on missing values as well as algorithm parameters) may not lead to an optimal solution. Moreover, errors made in learning distributions of missing features may undermine

the complete-data algorithm.

The other class of methods explicitly addresses missing values during the model-learning procedure. The work proposed by [CHE⁺08] represents a special case, in which no model is assumed for *structurally absent* values; the margin for the support vector machine (SVM) is re-scaled according to the observed features for each instance. Empirical results [CHE⁺08] show that this procedure is comparable to several single-imputation methods when values are *missing at random*. Another recent work [DHS08] handles the missing features inside the procedure of learning a support vector machine (SVM), without constraining the distribution of missing features to any specific class. The main concern is that this method can only handle missing features in the training data; however, in many applications one cannot control whether missing values occur in the training or test data.

A widely employed approach for handling missing values within the algorithm involves maximum likelihood (ML) estimation via expectation maximization (EM) [DLR77]. Besides the latent variables (*e.g.*, mixture component indicators), the missing features are also integrated out in the E-step so that the likelihood is maximized with respect to model parameters in the M-step. The main difficulty is that the integral in the E-step is analytically tractable only when an assumption is made on the distribution of the missing features. For example, the intractable integral is avoided by requiring the features to be discrete [Ibr90], or assuming a Gaussian mixture model (GMM) for the features [GJ94, LLC07a]. The discreteness requirement is often too restrictive, while the GMM assumption is mild since it is well known that a GMM can approximate arbitrary continuous distributions.

In [LLC07a] the authors proposed a *quadratically gated mixture of experts* (QGME) where the GMM is used to form the gating network, statistically partitioning the feature space into quadratic subregions. In each subregion, one linear classifier works as a local “expert”. As a mixture of experts [JJNH91], the QGME is capable of

addressing a classification problem with a nonlinear decision boundary in terms of multiple local experts; the simple form of this model makes it straightforward to handle incomplete data without completing kernel functions [Gra02, WC05]. However, as in many mixture-of-expert models [JJNH91, WR94, XJH95], the number of local experts in the QGME must be specified initially, and thus a model-selection stage is in general necessary. Moreover, since the expectation-maximization method renders a point (single) solution that maximizes the likelihood, over-fitting may occur when data are scarce relative to the model complexity. In this thesis we propose a non-parametric extension for the QGME model in this dissertation, which reserves its desirable properties while addressing the aforementioned two issues successfully.

1.5 Multi-task Learning

In traditional supervised learning for a single learning task, a sufficient quantity of training data is required for satisfying generalization ability. However, training data may be costly (e.g., due to time, cost or risk) to obtain in many applications, for example, undertaking a surgery to disclose whether a patient has a benign or malignant tumor and labeling a buried target such as a land mine or clutter by digging a hole and excavating it [ZCY⁺03].

One learning task may be decomposed into multiple correlated subtasks naturally. For example, in remote sensing, one may collect multiple sets of data at different geographical locations using the same sensor; in medical diagnoses, physical measurements of patients with a common disease may be collected from multiple countries; in image reconstruction, people usually take several images for the same target. In these examples, each location, country or image corresponds to a sub-task, respectively. Compared to learning each subtask individually, which is called

single-task learning (STL), sharing information across subtasks to enhance overall performance is more desirable. This way of learning multiple tasks simultaneously to improve generalization performance is referred to as *multi-task learning* (MTL) [Car97].

The multi-task learning setting implies the following mild assumptions: (i) the measurements in different tasks must have some kind of inter-relationship, so that the information contained in corresponding measurements is transferable; (ii) at least some of the tasks are related (dependent on each other), so that one task may be informative for learning another one; however, (iii) there exists difference between tasks, which means it is not appropriate to pool the data in all tasks together and treat them as a single task. The result of such cooperation of data from correlated tasks is that the “effective” data for each task are increased and therefore an overall better generalization performance could be expected, especially when the available data are scarce. Caruana [Car97] gave an overview of MTL and demonstrated it on multiple problems.

In recent years, especially over the last decade, multi-task learning has become an important topic in the machine learning community. The diversity of MTL approaches mainly comes from the various ways of information sharing among tasks. There are non-Bayesian methods, for example, kernel methods with regularization are extended to the case of multi-task learning by defining particular forms of kernels [EMP05]; under the frequentist setting, a common predictive structure was transferred from multiple tasks to the target problem [AZ05]. In the Bayesian framework, usually some hidden variables or the prior of hidden variables are shared. For example, for artificial neural networks, usually a common hidden “internal representation” was shared by tasks from the same environment [Bax95, Bax00]; the parameters of a covariance function for Gaussian process priors are shared across multiple tasks [LP04, YST05]; a hierarchical structure is favored by many researchers recently,

where the information is transferred by a common prior in hierarchical Bayesian models [YST⁺03, YTY04, STY05, ZGY06].

Hierarchical Bayesian models are an important class of statistical models that allow the flexibility to simultaneously model both individual tasks and the correlations between tasks. In the hierarchical structure, usually the bottom layer is for individual tasks with the same models as in the single-task learning; however, on the upper layer, a common prior is placed on those parameters that are assumed to transfer so that the individual tasks could be connected. In this manner, individual tasks are learned independently given the common prior, which accounts for the contribution of data from all tasks. As a result, through the common prior, the learning of a task is affected by training data from the other tasks as well as its own training data.

A common assumption in MTL work has been that all tasks are equally related to each other; nevertheless, this is not always the case in many real applications. Given such an assumption, for those tasks different from others, the information transferred from other tasks may be misleading instead of helpful. For the first time, Thrun and O’Sullivan [TO96] proposed the task-clustering (TC) algorithm with K-nearest neighbors, which assumed that some tasks are more similar to each other. By introducing a Dirichlet process (DP) prior as the common prior in hierarchical Bayesian models, similarities between the various tasks could be automatically identified [XLCK07]. Based on unlabeled data in a neighborhood of feature space, a semi-supervised multi-task learning approach is proposed [LLC07c], where the common prior is quite similar to the DP. Instead of sharing every dimension of predictors in the same manner across tasks, the matrix stick-breaking process [XDC07] is developed as an extension of DP, which allows for sharing a part of predictors. Based on the hierarchical Dirichlet process (HDP) [TMIJB06], another extension of the Dirichlet process, we allow for local sharing in predictor space across tasks.

1.6 Collaborative Filtering

Other than the prediction of associated responses for new samples, the completion of the missing data itself may be of particular interest in various applications. When the data matrix involves multiple agents, viewpoints, or data sources, the missing values could be inferred by *collaborative filtering* (CF). Although collaborative filtering methods have been applied to many different kinds of data, user-rating data has become a recent focus (see for example [SRJ05, ABEV09, SM07, SM08, LU09, YLZG09]) and typically involves very large data sets with a large fraction of the elements missing, and to be inferred. In the context of user-preference data, taste information from many users (collaborating) is collected to make predictions (filtering) about the interests of some users. With an underlying assumption that the tastes of a user remain the same, the unknown (missing) preferences on items could be inferred from the given (observed) preferences. For example, collaborative filtering for movie tastes could make predictions about which movies a user should like given a partial list of that user's tastes (likes or dislikes).

Early CF approaches are *memory-based*, where user ratings are used to compute similarity between users or items, based on which predictions are further made. Neighborhood-based CF is a non-parametric (of the first meaning) example [SK09]. This class of approaches are easy to implement, and new data can be added easily and incrementally. However, these methods usually involve calculating Pearson correlations between all the user pairs or item pairs, which prevents the scalability to large data sets; and the similarities calculated from common items may be unreliable when data are sparse (common items tend to be few) [SK09].

Along another line of research, models are developed to find patterns based on training data, and used to make predictions for test data. There are many so-called

model-based CF algorithms [SRJ05, CR09, ABEV09, SM07, SM08, LU09, YLZG09], with the goal of uncovering latent factors. This class of methods discard the data for training and make prediction based on models summarized from the training data. This helps scale up the algorithms to large data sets and improves the prediction performance. However, learning a model is more expensive than simply memorizing users’ preferences. One needs to make a tradeoff between prediction performance, scalability and computational expenses.

Consider a data matrix $\mathbf{Y} \in \mathfrak{R}^{I \times J}$, e.g., a rating matrix for I users on J movies. Many of CF models are based on matrix factorization, i.e., to fit the target data matrix \mathbf{Y} with a factorable matrix $\mathbf{X} = \mathbf{UV}^T$, where $\mathbf{U} \in \mathfrak{R}^{I \times D}$ and $\mathbf{V} \in \mathfrak{R}^{J \times D}$ with D the dimensionality of latent factor space. Some approaches address this problem from the standpoint of non-Bayesian optimization [SRJ05, CR09, Hof04, LS99, SJ01]: fitting a target matrix \mathbf{Y} with a structure-constrained matrix \mathbf{X} by minimizing error with some loss function. A simple example of a technique for finding such a factorization is principal components analysis (PCA), minimizing sum-of-squares error. Various constraints on \mathbf{U} and \mathbf{V} have been considered [LS99, SJ01, SRJ05, CR09]. For example, low-rank approximations constrain the dimensionality D for \mathbf{U} and \mathbf{V} [SRJ05, CR09]; sparsity constraint requires the number of nonzero components in \mathbf{U} or \mathbf{V} is small [SJ01]; non-negativity has also been suggested for better capturing the structure in \mathbf{Y} [SBHM09]. Recently, some work has been viewing this problem from a Bayesian perspective [SM07, SM08, LU09, YLZG09]: an error matrix is introduced such that $\mathbf{Y} = \mathbf{UV}^T + \mathbf{E}$, and priors are imposed on \mathbf{U} , \mathbf{V} and \mathbf{E} . With i.i.d. Gaussian noise, columns of \mathbf{U} and \mathbf{V} are both assumed to be multi-variant Gaussian in probabilistic matrix factorization (PMF) [SM07]. Bayesian probabilistic matrix factorization (BPMF) [SM08] provides a fully Bayesian treatment for the PMF, and may be viewed as Bayesian principal component analysis (BPCA) [Bis99]. In some other methods, the Gaussian process (GP) is applied in different ways [LU09, YLZG09] to

introduce nonlinearity or contiguity.

A challenge is the need to estimate or define D , the dimensionality of the space in which the matrix lives. For example, in BPMF the matrices \mathbf{U} and \mathbf{V} define a (typically low-dimensional) linear subspace in which \mathbf{X} resides. In [SRJ05, SM08] regularizers/priors are employed to constrain the dimensionality D . Lawrence [LU09] provides a nonlinear generalization by using a Gaussian process (GP) based formulation; for this model there is a related latent-space dimension in the GP kernel that needs to be tuned (*e.g.*, via cross-validation) for different data sets and different missing ratios.

Sometimes information other than the matrix data is available, such as the characteristics for users (*e.g.*, age, occupation, gender) and movies (*e.g.*, genre, year, director). Another aspect of work that is gaining recent attention is the use of such auxiliary information, beyond the matrix data, when performing inference [ABEV09, YLZG09].

1.7 Summary of the Remaining Chapters

Chapter 2 provides background on the Dirichlet process [Fer73] and its extensions, which are employed in Chapter 3 and 4. We start from its definition and two practical representations: the Pólya urn scheme and the stick-breaking construction. Dirichlet process mixtures are then presented as a typical way of applying the DP for clustering data. After that, the hierarchical Dirichlet process is introduced as an extension for hierarchical data clustering.

In Chapter 3, we form the proposed classifier capable of dealing missing features by extending the finite QGME [LLC07a] to an infinite QGME (iQGME) via a Dirichlet process (DP) [Fer73] prior. A variant is proposed to handle high dimen-

sional data sets. A mean-field variational Bayesian algorithm is developed for model inference. The proposed model is first illustrated by a synthetic data set, and then demonstrated on benchmark data sets and two real applications with natural missing values.

A further extension to multi-task setting via a hierarchical Dirichlet process is made in Chapter 4. Experimental results are reported to demonstrate the proposed multi-task classifier on a landmine detection problem and a handwritten-letter classification problem.

Chapter 5 presents a novel matrix-completion method, Bayesian singular value decomposition (BSVD), which is inferred using a Gibbs sampler. We generalize the basic BSVD model to make use of auxiliary information and handle counting data via a probit regression link. An active learning procedure is developed based on “error bars” on the inferred matrix values provided by the BSVD model. Experiments results are reported on synthetic matrices, a benchmark movie-rating data set, and a newly introduced Major League Baseball (MLB) data from 1954-2008.

Chapter 6 concludes the dissertation with a review of the main contributions and limitations. We also discuss research directions for future work.

Dirichlet Process and Its Extensions

In this chapter, we provide a brief review of the Dirichlet process (DP) [Fer73], which is an important technique implemented in Chapters 3 and 4. We first give the mathematical definition of the DP [Fer73] and then introduce the Pólya urn scheme [BM73] and the stick-breaking construction [Set94], which are two practical representations of the DP. The discrete nature of the DP leads to a clustering property which is attractive for mixture modeling. Such Dirichlet process mixtures (DPM) are formulated and discussed as a typical application of the DP. As an extension the hierarchical Dirichlet process [TMIJB06] is presented and will be implemented in Chapter 4.

2.1 Dirichlet Process

The Dirichlet process is a random process originally analyzed by Ferguson [Fer73]. Assume \mathcal{S} is a set, over which a Borel σ -algebra \mathcal{B} is defined. Let G_0 be a probability measure on $(\mathcal{S}, \mathcal{B})$, and let α be a positive real number. The distribution of a probability measure G is defined as a Dirichlet process on $(\mathcal{S}, \mathcal{B})$ with base distribution G_0 and precision parameter α if for any finite measurable partition of \mathcal{S} , denoted

by A_1, \dots, A_K , the joint distribution of $(G(A_1), \dots, G(A_K))$ is a finite-dimensional Dirichlet distribution with parameters $(G_0(A_1), \dots, G_0(A_K))$, i.e.,

$$(G(A_1), \dots, G(A_K)) \sim \text{Dir}(G_0(A_1), \dots, G_0(A_K)).$$

Based on the above definition, Ferguson[Fer73] proved that the posterior of G is still a Dirichlet process, with an updated base distribution and precision, which is a desirable property for Bayesian inference. Nevertheless, this mathematical definition does not lead to a data-generating process for the DP directly. Two approaches to generating i.i.d. samples from G , a DP draw, are introduced below. One is the Pólya urn scheme [BM73], where G is marginalized and the i.i.d. samples are generated from conditional distributions; the other is the stick-breaking construction, which constitutes G explicitly.

2.1.1 Pólya urn scheme

Assuming $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n$ is a sample of size n drawn iid from G , the mathematical representation of the Dirichlet process model is

$$\begin{aligned} \boldsymbol{\theta}_i | G &\stackrel{iid}{\sim} G, \\ G &\sim \mathcal{DP}(\alpha, G_0), \end{aligned}$$

where $i = 1, \dots, n$.

After integrating out G , we obtain the conditional distribution of $\boldsymbol{\theta}_r$ given the other $n - 1$ values as

$$p(\boldsymbol{\theta}_r | \boldsymbol{\theta}_{-r}, \alpha, G_0) = \frac{\alpha}{\alpha + n - 1} G_0(\boldsymbol{\theta}_r) + \frac{1}{\alpha + n - 1} \sum_{i \neq r}^n \delta_{\boldsymbol{\theta}_i},$$

where $\boldsymbol{\theta}_{-r} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{r-1}, \boldsymbol{\theta}_{r+1}, \dots, \boldsymbol{\theta}_n)$ and $\delta_{\boldsymbol{\theta}_i}$ is a unit point measure concentrated at $\boldsymbol{\theta}_i$.

If we denote the K distinct values among $\boldsymbol{\theta}_{-r}$ as $\boldsymbol{\theta}_k^*$, $k = 1, \dots, K$, the conditional distribution can be alternatively written as

$$p(\boldsymbol{\theta}_r | \boldsymbol{\theta}_{-r}, \alpha, G_0) = \frac{\alpha}{\alpha + n - 1} G_0(\boldsymbol{\theta}_r) + \frac{1}{\alpha + n - 1} \sum_{k=1}^K N_{-r,k} \delta_{\boldsymbol{\theta}_k^*}, \quad (2.1)$$

where $N_{-r,k}$ denotes the number of instances equal to the k th distinct value $\boldsymbol{\theta}_k^*$ among the $n - 1$ instances other than $\boldsymbol{\theta}_r$. From (2.1), we can see:

1. Since there exists a finite positive probability that two instances are of exactly the same value, draws from a Dirichlet process are discrete measures with probability one.
2. The Dirichlet process exhibits a property of self-reinforcing by counting, i.e., the more often a given value $\boldsymbol{\theta}_k^*$ has been sampled in the past, the more likely it is to be sampled in the future. This self-reinforcing mechanism leads to the tendency of clustering.

The process of drawing instances according to (2.1) is usually referred to as Pólya urn scheme [BM73]. Specifically, consider an urn that is empty at the beginning. After a color is randomly picked from the spectrum according to some base distribution, one ball of this color is put into the urn. After that, at each step, we either (i) draw a ball from the urn and put it back with another ball of the same color, or (ii) place a ball of a randomly selected new color into the urn. If we define the base distribution over the color spectrum as G_0 and the probability of performing (ii) with n balls in the urn as $\frac{\alpha}{\alpha+n}$, this process is exactly what (2.1) describes.

Since a new ball (instance) tends to have the same color (distinct value / cluster) with a large population, the Dirichlet process has an implicit mechanism of clustering instances into groups. The probability of creating a new cluster is controlled by the parameter α , and a larger α encourages more clusters. In the limit of $\alpha \rightarrow \infty$, only

(ii) is performed at each step, and as a result, every instance θ_i is distinct. As a sample from G , $\{\theta_i\}_{i=1}^n$ are guaranteed to be in the support of G_0 ; therefore, we have the following statement in the limiting case

$$\lim_{\alpha \rightarrow \infty} G = G_0.$$

On the other hand, in the limit of $\alpha \rightarrow 0$, the step (i) is always repeated and hence all the balls are in the same cluster as the first ball. In this limit,

$$\lim_{\alpha \rightarrow 0} G = \delta_{\theta_1}.$$

In addition, it can be shown that $E[G] = G_0$, which means that the base distribution represents our expectation concerning G .

From above, the Pólya urn scheme provides an intuitive way to understand the mechanism of the DP; however, we also noticed that an explicit form of G is absent in this representation.

2.1.2 Stick-breaking construction

The stick-breaking construction [Set94] provides an explicit form of G , which is desirable in some situation. Specifically, assume a DP prior with base measure G_0 and precision parameter α is assigned on a measure G . It has been proven by Sethuraman [Set94] that G may be constructed as

$$G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h^*}, \text{ with } 0 \leq \pi_h \leq 1 \text{ and } \sum_{h=1}^{\infty} \pi_h = 1 \text{ a.s.}, \quad (2.2)$$

where

$$\pi_h = V_h \prod_{l=1}^{h-1} (1 - V_l), \quad V_h \stackrel{iid}{\sim} \text{Beta}(1, \alpha), \quad \theta_h^* \stackrel{iid}{\sim} G_0.$$

From (2.2), it is clear that G is actually a probability mass function (pmf) with an infinite set of weights $\{\pi_h\}_{h=1}^\infty$ at infinite atoms $\{\theta_h^*\}_{h=1}^\infty$.

According to (2.2), the construction mechanism of weights and atoms may be described as following. Assume that a “whole” stick of unit-length is to be broken into pieces. At the first step ($h = 1$), a sample V_1 is drawn from $\text{Beta}(1, \alpha)$ and taken as the length of the first piece (stick) π_1 , and simultaneously a sample θ_1^* is drawn from the base distribution G_0 . After that, at step h , together with a sample θ_h^* drawn from G_0 , a fraction given by V_h is cut from the remaining part of step $h - 1$ and taken as the h th stick π_h .

This stick-breaking procedure may be truncated in practice. Let us say we decide to stop at some step N by keeping the remaining length of step $N - 1$ as the last stick π_N , which means V_N is forced to be one instead of a draw from the beta distribution. This yields an N -level truncation approximation to a draw from the Dirichlet process,

$$G = \sum_{h=1}^N \pi_h \delta_{\theta_h^*}. \quad (2.3)$$

To address the issue of setting a truncation level, Ishwaran and James [IJ01] prove theorems for selecting an appropriate truncation level N , which results in a model that approximates the probability mass function with an infinite number of atoms well. Furthermore, Papaspiliopoulos and Roberts [PR08] proposed a retrospective sampling scheme, which adaptively selects the number of necessary sticks, and thus avoids truncation.

As the parameter of the beta distribution, from which independent $\{V_h\}_{h=1}^\infty$ are sampled, α affects the number of large sticks and thus the appropriate truncation level. For draws from a beta distribution, $E[V_h] = \frac{1}{1+\alpha}$ and straightforwardly $E[\pi_h] = \frac{\alpha^{h-1}}{(1+\alpha)^h}$. When α is rather small, $E[\pi_h] \approx \alpha^{h-1}$, which exponentially decays with h , and hence only the first few sticks may have large weights (encouraging a small

number of clusters); when α is very large, $E[\pi_h] \approx \frac{1}{1+\alpha}$, which means that all sticks have similar weights (encouraging many small clusters). Clearly, the influence of α on the stick-breaking process coincides with that in Pólya urn scheme.

2.2 Dirichlet Process Mixtures

The representation in (2.2) shows that draws from a DP are discrete with probability one. This discrete nature makes the DP suitable for the mixture-modeling (data-partition) problem. The idea is basically to associate a mixture component with each atom in G . Specifically, *Dirichlet process mixtures* (DPM) are constructed as follows. Consider a set of data $\mathbf{x}_1, \dots, \mathbf{x}_n$ with an underlying model $\mathbf{x}_i \sim g(\boldsymbol{\theta}_i)$, where the parameters, $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n$, are drawn identically independently from a prior distribution G , which itself is sampled from a Dirichlet process $\mathcal{DP}(\alpha, G_0)$. In a mathematical way, the hierarchical model of the DPM is

$$\begin{aligned} \mathbf{x}_i &\sim g(\boldsymbol{\theta}_i), i = 1, \dots, n, \\ \boldsymbol{\theta}_i | G &\stackrel{iid}{\sim} G, i = 1, \dots, n, \\ G &\sim \mathcal{DP}(\alpha, G_0). \end{aligned}$$

Usually, data $\mathbf{x}_1, \dots, \mathbf{x}_n$ are observable but parameters $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n$ are hidden; with a DP prior measure, which is discrete, two or more parameters may share the same value as we discussed before. As a result, those data points associated with the same value could be clustered together and the number of clusters is data-driven.

Dirichlet process mixtures provide an alternative to methods that attempt to select a particular number of mixture components, for example, Gaussian mixture model (GMM), hidden Markov model (HMM) and mixture of experts (ME). Much work has been done on such Dirichlet process mixture models [Ant74, EW95, MM98, QPC07, DPP07, MEW96, RG02, MO06, SN09, RDG09, HBP10]. In Chapter 3 we

develop a novel classifier using Dirichlet process mixtures priors, and review related previous work in detail.

2.3 Hierarchical Dirichlet Process

The Dirichlet process has provided a solution for data partition or clustering in one group. Teh *et al.* extended the DP to the hierarchical Dirichlet process (HDP) [TMIJB06] for the setting in which the data are subdivided into a number of groups. Given the goal of solving a clustering problem within each group, a random measure G_j may be assumed to be associated with each task j , where each G_j is a draw from a group-specific Dirichlet process $\mathcal{DP}(\alpha_j G_{j0})$. A natural way to make connections between these clustering problems is to share the same group-specific DPs, i.e., $G_j \sim \mathcal{DP}(\alpha G_0)$. In order to allow for sharing of clusters between groups, the authors forced the common base measure G_0 to be discrete by considering G_0 itself a draw from an upper-level Dirichlet process $\mathcal{DP}(\beta H)$, i.e.,

$$\begin{aligned} G_j &\sim \mathcal{DP}(\alpha G_0), \text{ for } j = 1, \dots, J, \\ G_0 &\sim \mathcal{DP}(\beta H). \end{aligned}$$

As a draw from a Dirichlet process, G_0 is discrete with probability one and has a stick-breaking representation as in (2.2). With such a base measure, the group-dependent DPs reuse the atoms θ_h^* defined in G_0 , yielding the desired sharing of atoms among groups.

The hierarchical Dirichlet process has been implemented in various applications, such as sequential data analysis [NCD07], linguistic modeling [LPJK07] and target tracking [EBFW07]. In Chapter 4 a multi-task classification model allowing for local sharing in feature space between the tasks will be developed via the HDP prior.

Classification with Incomplete Data

3.1 Introduction

We address single-task learning problems in this chapter by extending the quadratically gated mixture of experts (QGME) [LLC07a] to a fully Bayesian setting, with the number of local experts inferred automatically via a non-parametric DP prior.

The Dirichlet process [Fer73] has been an active topic in many applications since the middle 1990s, for example, density estimation [EW95, MM98, DPP07] and regression/curve fitting/classification [MEW96, RG02, MO06, WACD09, SN09, RDG09, HBP10]. The latter group is relevant to classification problems of interest in this chapter. The work in [MEW96] jointly modeled inputs and responses as a Dirichlet process mixture of multivariate normals, while [RDG09] extended this model to simultaneously estimate multiple curves using dependent DP. In [RG02] and [MO06] two approaches to constructing infinite mixtures of Gaussian Process (GP) experts were proposed. The difference is that [MO06] specified the gating network using a multivariate Gaussian mixture instead of a (fixed) input-dependent Dirichlet Process. In [WACD09] the gating network is defined based on the kernel stick-breaking pro-

cess (KSBP) [DP08], which provides an alternative way to construct input-dependent DP; the experts are defined as probability mass functions over classes. In [SN09] another form of infinite mixtures of experts was proposed, where experts are specified by a multinomial logit (MNL) model (also called softmax) and the gating network is Gaussian mixture model with independent covariates. Further, [HBP10] generalized existing DP-based nonparametric regression models to accommodate different types of covariates and responses, and further gave theoretical guarantees for this class of models.

Our focus in this chapter is on developing classification models that handle incomplete inputs/covariates efficiently using Dirichlet process. Some of the above Dirichlet process regression models are potentially capable of handling incomplete inputs/features; however, none of them actually deal with such problems. In [MEW96], although the joint multivariate normal assumption over inputs and responses endow this approach with the potential of handling missing features and/or missing responses naturally, a good estimation for the joint distribution does not guarantee a good estimation for classification boundaries. Other than a full joint Gaussian distribution assumption, explicit classifiers were used to model the conditional distribution of responses given covariates in the models proposed in [MO06] and [SN09]. These two models are highly related to the infinite quadratically gated mixture of experts (iQGME) model we propose in this chapter. The independence assumption of covariates in [SN09] leads to efficient computation but is not appealing for handling missing features. With Gaussian process experts [MO06], the inference for missing features is not analytical for fast inference algorithms such as expectation maximization (EM) [DLR77] and variational Bayesian [Bea03], and the computation could be prohibitive for large data sets. The iQGME seeks a balance between the ease of inference, computational burden and the ability of handling missing features.

For high-dimensional data sets, we develop a variant of our model based on mix-

tures of factor analyzers (MFA) [GH96, GB00], where a low-rank assumption is made for the covariance matrices of high-dimensional inputs in each cluster. Throughout, efficient inference is implemented via the variational Bayesian (VB) method [Bea03]. To quantify the accuracy of the VB results, we also perform comparative studies based on Gibbs sampling.

3.2 Infinite Quadratically Gated Mixture of Experts

3.2.1 Quadratically gated mixture of experts

Consider a binary classification problem with real-valued P -dimensional column feature vectors \mathbf{x}_i and corresponding class labels $y_i \in \{1, -1\}$. We assume binary labels for simplicity, while the proposed method may be directly extended to cases with more than two classes. Latent variables t_i are introduced as “soft labels” associated with y_i , as in probit models [AC93], where $y_i = 1$ if $t_i > 0$ and $y_i = -1$ if $t_i \leq 0$. The finite quadratically gated mixture of experts (QGME) [LLC07a] is defined as

$$(t_i | z_i = h) \sim \mathcal{N}(\mathbf{w}_h^T \mathbf{x}_i^b, 1), \quad (3.1)$$

$$(\mathbf{x}_i | z_i = h) \sim \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}), \quad (3.2)$$

$$(z_i | \boldsymbol{\pi}) \sim \sum_{h=1}^K \pi_h \delta_h, \quad (3.3)$$

with $\sum_{h=1}^K \pi_h = 1$, and where δ_h is a point measure concentrated at h (with probability one, a draw from δ_h will be h). The $(P+1) \times K$ matrix \mathbf{W} has columns \mathbf{w}_h , where each \mathbf{w}_h are the weights on a local linear classifier, and the \mathbf{x}_i^b are feature vectors with an intercept, *i.e.*, $\mathbf{x}_i^b = [\mathbf{x}_i^T, 1]^T$. A total of K groups of \mathbf{w}_h are introduced to parameterize the K experts. With probability π_h the indicator for the i th data point satisfies $z_i = h$, which means the h th local expert is selected, and \mathbf{x}_i is distributed according to a P -variate Gaussian distribution with mean $\boldsymbol{\mu}_h$ and precision $\boldsymbol{\Lambda}_h$.

It can be seen that the QGME is highly related to the mixture of experts (ME) [JJNH91] and the hierarchical mixture of experts (HME) [JJ94] if we write the conditional distribution of labels as

$$p(y_i|\mathbf{x}_i) = \sum_{h=1}^K p(z_i = h|\mathbf{x}_i)p(y_i|z_i = h, \mathbf{x}_i), \quad (3.4)$$

where

$$p(y_i|z_i = h, \mathbf{x}_i) = \int_{t_i y_i > 0} \mathcal{N}(t_i | \mathbf{w}_h^T \mathbf{x}_i, 1) dt_i, \quad (3.5)$$

$$p(z_i = h|\mathbf{x}_i) = \frac{\pi_h \mathcal{N}_P(\mathbf{x}_i | \boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})}{\sum_{k=1}^K \pi_k \mathcal{N}_P(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})}. \quad (3.6)$$

From (3.4), as a special case of the ME, the QGME is capable of handling nonlinear problems with linear experts characterized in (3.5). However, unlike other ME models, the QGME probabilistically partitions the feature space through a mixture of K Gaussian distributions for \mathbf{x}_i as in (3.6). This assumption on the distribution of \mathbf{x}_i is mild since it is well known that a Gaussian mixture model (GMM) is general enough to approximate any continuous distribution. In the QGME, \mathbf{x}_i as well as y_i are treated as random variables (generative model) and we consider a joint probability $p(y_i, \mathbf{x}_i)$ instead of a conditional probability $p(y_i|\mathbf{x}_i)$ for fixed \mathbf{x}_i as in most ME models (which are typically discriminative). Previous work on the comparison between discriminative and generative models may be found in [NJ02, LJ08]. In the QGME, the GMM of the inputs \mathbf{x}_i plays two important roles: *i*) as a gating network, while *ii*) enabling analytic incorporation of incomplete data during classifier inference (as discussed further below).

The QGME [LLC07a] is inferred via the expectation-maximization (EM) method, which renders a point-estimate solution for an initially specified model (3.1)-(3.3), with a fixed number K of local experts. Since learning the correct model requires

model selection, and moreover in many applications there may exist no such fixed “correct” model, in the work reported here we infer the full posterior for a QGME model with the number of experts data-driven. The objective can be achieved by imposing a nonparametric Dirichlet process (DP) prior.

3.2.2 Infinite QGME via Dirichlet process

Consider a classification task with a training data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) : i = 1, \dots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^P$ and $y_i \in \{-1, 1\}$. With soft labels t_i introduced as in Section 3.2.1, the infinite QGME (iQGME) model is achieved via a DP prior imposed on the measure G of $(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, \mathbf{w}_i)$, the hidden variables characterizing the density function of each data point (\mathbf{x}_i, t_i) . For simplicity, the same symbols are used to denote parameters associated with each data point and the distinct values, with subscripts i and h indexing data points and unique values, respectively:

$$\begin{aligned} (\mathbf{x}_i, t_i) &\sim \mathcal{N}_P(\mathbf{x}_i | \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1}) \mathcal{N}(t_i | \mathbf{w}_i^T \mathbf{x}_i, 1), \\ (\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, \mathbf{w}_i) &\stackrel{iid}{\sim} G, \\ G &\sim \mathcal{DP}(\alpha G_0), \end{aligned} \tag{3.7}$$

where the base measure G_0 is factorized as the product of a normal-Wishart prior for $(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h)$ and a normal prior for \mathbf{w}_h , for the sake of conjugacy. As discussed in Chapter 2, data samples cluster automatically, and the same mean $\boldsymbol{\mu}_h$, covariance matrix $\boldsymbol{\Lambda}_h$ and regression coefficients (expert) \mathbf{w}_h are shared for a given cluster h . Using the stick-breaking construction, we elaborate (3.7) as follows for $i = 1, \dots, n$

and $h = 1, \dots, \infty$:

Data generation:

$$\begin{aligned} (t_i | z_i = h) &\sim \mathcal{N}(\mathbf{w}_h^T \mathbf{x}_i^b, 1), \\ (\mathbf{x}_i | z_i = h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}), \end{aligned}$$

Drawing indicators:

$$\begin{aligned} z_i &\sim \sum_{h=1}^{\infty} \pi_h \delta_h, \quad \text{where } \pi_h = V_h \prod_{l < h} (1 - V_l), \\ V_h &\sim \text{Be}(1, \alpha), \end{aligned}$$

Drawing parameters from G_0 :

$$\begin{aligned} (\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_h | \mathbf{m}_0, u_0^{-1} \boldsymbol{\Lambda}_h^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_h | \mathbf{B}_0, \nu_0), \\ \mathbf{w}_h &\sim \mathcal{N}_{P+1}(\boldsymbol{\zeta}, [\text{diag}(\boldsymbol{\lambda})]^{-1}), \quad \text{where } \boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{P+1}]. \end{aligned}$$

Furthermore, to achieve a robust algorithm, we assign diffuse hyper-priors on several crucial parameters. As discussed in Chapter 2, the scaling parameter α reflects our prior belief on the number of clusters. For the sake of conjugacy, a diffuse Gamma prior is usually assumed for α as suggested by [WME94]. In addition, parameters $\boldsymbol{\zeta}, \boldsymbol{\lambda}$ characterizing the prior of the distinct local classifiers \mathbf{w}_h are another set of important parameters, since we focus on classification tasks. Normal-Gamma priors are the conjugate priors for the mean and precision of a normal density. Therefore,

$$\begin{aligned} \alpha &\sim \text{Ga}(\tau_{10}, \tau_{20}), \\ (\boldsymbol{\zeta} | \boldsymbol{\lambda}) &\sim \mathcal{N}_{P+1}(\mathbf{0}, \gamma_0^{-1} [\text{diag}(\boldsymbol{\lambda})]^{-1}), \\ \lambda_p &\sim \text{Ga}(a_0, b_0), \quad p = 1, \dots, P + 1, \end{aligned}$$

where $\tau_{10}, \tau_{20}, a_0, b_0$ are usually set to be much less than one and of about the same magnitude, so that the constructed Gamma distributions with means about one and large variances are diffuse; γ_0 is usually set to be around one.

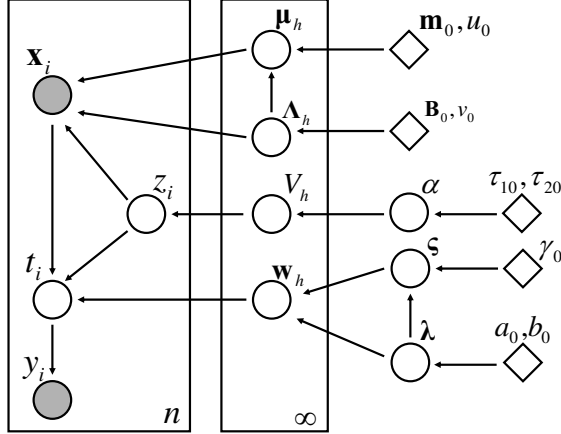


FIGURE 3.1: Graphical representation of the iQGME for single-task learning. All circles denote random variables, with shaded ones indicating observable data, and bright ones representing hidden variables. Diamonds denote fixed hyper-parameters, boxes represent independent replicates with the numbers of copies indicated at the lower-right corner, and arrows indicate the dependence between variables (pointing from parents to children).

The graphical representation of the iQGME for single-task learning is shown in Figure 3.1. We notice that a possible variant with sparse local classifiers could be obtained if we impose zero mean for the local classifiers \mathbf{w}_h , *i.e.*, $\boldsymbol{\zeta} = \mathbf{0}$, and retain the Gamma hyper-prior for the precision $\boldsymbol{\lambda}$, as in the relevance vector machine (RVM) [Tip00], which employs a corresponding Student-t sparseness prior on the weights. Although this sparseness prior is useful for seeking relevant features in many applications, imposing the same sparse pattern for all the local experts is not desirable.

3.2.3 A variant for high-dimensional problems

For the classification problem, we assume access to a training data set $\mathcal{D} = \{(\mathbf{x}_i, y_i) : i = 1, \dots, n\}$, where feature vectors $\mathbf{x}_i \in \mathbb{R}^P$ and labels $y_i \in \{-1, 1\}$. We have assumed that the feature vectors of objects in cluster h are generated from a P -variate normal distribution with mean $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Lambda}_h^{-1}$, *i.e.*,

$$(\mathbf{x}_i | z_i = h) \sim \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}) \quad (3.8)$$

It is well known that each covariance matrix has $P(P + 1)/2$ parameters to be estimated. Without any further assumption, the estimation of these parameters could be computationally prohibitive for large P , especially when the number of available training data n is small, which is common for classification applications. By imposing an approximately low-rank constraint on the covariances, as in well-studied mixtures of factor analyzers (MFA) models [GH96, GB00], the number of unknowns could be significantly reduced. Specifically, assume a vector of standard normal latent factors $\mathbf{s}_i \in \mathbb{R}^{T \times 1}$ for data \mathbf{x}_i , a factor loading matrix $\mathbf{A}_h \in \mathbb{R}^{P \times T}$ for cluster h , and Gaussian residues $\boldsymbol{\epsilon}_i$ with diagonal covariance matrix $\psi_h \mathbf{I}_P$, then

$$(\mathbf{x}_i | z_i = h) \sim \mathcal{N}_P(\mathbf{A}_h \mathbf{s}_i + \boldsymbol{\mu}_h, \psi_h^{-1} \mathbf{I}_P).$$

Marginalizing \mathbf{s}_i with $\mathbf{s}_i \sim \mathcal{N}_T(\mathbf{0}, \mathbf{I}_T)$, we recover (3.8), with $\boldsymbol{\Lambda}_h^{-1} = \mathbf{A}_h \mathbf{A}_h^T + \psi_h^{-1} \mathbf{I}_P$. The number of free parameters is significantly reduced if $T \ll P$.

In this thesis, we modify the MFA model for classification applications with scarce samples. First, we consider a common loading matrix \mathbf{A} for all the clusters, and introduce a binary vector \mathbf{b}_h for each cluster to select which columns of \mathbf{A} are used, *i.e.*,

$$(\mathbf{x}_i | z_i = h) \sim \mathcal{N}_P(\mathbf{A} \text{diag}(\mathbf{d} \circ \mathbf{b}_h) \mathbf{s}_i + \boldsymbol{\mu}_h, \psi_h^{-1} \mathbf{I}_P), \quad (3.9)$$

where each column of \mathbf{A} , $\mathbf{A}_t \sim \mathcal{N}_P(\mathbf{0}, P^{-1} \mathbf{I}_P)$, $\mathbf{s}_i \sim \mathcal{N}_T(\mathbf{0}, \mathbf{I}_T)$, \mathbf{d} is a vector responsible for scale, and \circ is a component-wise (Hadamard) product. For \mathbf{d} we employ the prior $d_t \sim \mathcal{N}(0, \beta_t^{-1})$ with $\beta_t \sim \text{Gam}(c_0, d_0)$. Furthermore, we let the algorithm infer the intrinsic number of factors by imposing a low-rank belief for each cluster through the prior of \mathbf{b}_h , *i.e.*,

$$b_{h,t} \sim \text{Bern}(\pi_{h,t}), \quad \pi_{h,t} \sim \text{Beta}(a_0/L, b_0(L - 1)/L),$$

where L is a large number, which defines the largest possible dimensionality the algorithm may infer. Through the choice of a_0 and b_0 we impose our prior belief

about the intrinsic dimensionality of cluster h (upon integrating out the draw $\boldsymbol{\pi}_h$, the number of non-zero components of \mathbf{b}_h is drawn from $\text{Binomial}[L, a_0/(a_0+b_0(L-1))]$). As a result, both the number of clusters and the dimensionality of each cluster is inferred by this variant of iQGME.

With this form of iQGME, we could build local linear classifiers in either the original feature space or the (low-dimensional) space of latent factors \mathbf{s}_i . For the sake of computational simplicity, we choose to classify in the low-dimensional factor space.

3.3 Incomplete Data Problem

In the above discussion it was assumed that all components of the feature vectors were available (no missing data). In this section, we consider the situation for which feature vectors \mathbf{x}_i are partially observed. We partition each feature vector \mathbf{x}_i into observed and missing parts, $\mathbf{x}_i = [\mathbf{x}_i^{o_i}; \mathbf{x}_i^{m_i}]$, where $\mathbf{x}_i^{o_i} = \{x_{ip} : p \in o_i\}$ denotes the subvector of observed features and $\mathbf{x}_i^{m_i} = \{x_{ip} : p \in m_i\}$ represents the subvector of missing features, with o_i and m_i denoting the set of indices for observed and missing features, respectively. Each \mathbf{x}_i has its own observed set o_i and missing set m_i , which may be different for each i . Following a generic notation [SG02], we refer to \mathbf{R} as the missingness. For an arbitrary missing pattern, \mathbf{R} could be defined as a missing data indicator matrix, that is,

$$R_{ip} = \begin{cases} 1, & x_{ip} \text{ observed,} \\ 0, & x_{ip} \text{ missing.} \end{cases}$$

We use $\boldsymbol{\xi}$ to denote parameters characterizing the distribution of \mathbf{R} , which is usually called the *missing mechanism*. In the classification context, the joint distribution of class labels, observed features and the missingness \mathbf{R} may be given by integrating

out the missing features \mathbf{x}^m ,

$$p(y, \mathbf{x}^o, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\xi}) = \int p(y, \mathbf{x}|\boldsymbol{\theta})p(\mathbf{R}|\mathbf{x}, \boldsymbol{\xi})d\mathbf{x}^m. \quad (3.10)$$

To handle such a problem analytically, assumptions must be made on the distribution of \mathbf{R} . If the *missing mechanism* is conditionally independent of missing values \mathbf{x}^m given the observed data, *i.e.*, $p(\mathbf{R}|\mathbf{x}, \boldsymbol{\xi}) = p(\mathbf{R}|\mathbf{x}^o, \boldsymbol{\xi})$, the missing data are defined to be *missing at random* (MAR) [Rub76]. Consequently, (3.10) reduces to

$$p(y, \mathbf{x}^o, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\xi}) = p(\mathbf{R}|\mathbf{x}^o, \boldsymbol{\xi}) \int p(y, \mathbf{x}|\boldsymbol{\theta})d\mathbf{x}^m = p(\mathbf{R}|\mathbf{x}^o, \boldsymbol{\xi})p(y, \mathbf{x}^o|\boldsymbol{\theta}). \quad (3.11)$$

According to (3.11), the likelihood is factorizable under the assumption of MAR. As long as the prior $p(\boldsymbol{\theta}, \boldsymbol{\xi}) = p(\boldsymbol{\theta})p(\boldsymbol{\xi})$ (factorizable), the posterior

$$p(\boldsymbol{\theta}, \boldsymbol{\xi}|y, \mathbf{x}^o, \mathbf{R}) \propto p(y, \mathbf{x}^o, \mathbf{R}|\boldsymbol{\theta}, \boldsymbol{\xi})p(\boldsymbol{\theta}, \boldsymbol{\xi}) = p(\mathbf{R}|\mathbf{x}^o, \boldsymbol{\xi})p(\boldsymbol{\xi})p(y, \mathbf{x}^o|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

is also factorizable. For the purpose of inferring model parameters $\boldsymbol{\theta}$, no explicit specification is necessary on the distribution of the missingness. As an important special case of MAR, *missing completely at random* (MCAR) occurs if we can further assume that $p(\mathbf{R}|\mathbf{x}, \boldsymbol{\xi}) = p(\mathbf{R}|\boldsymbol{\xi})$, which means the distribution of missingness is independent of observed values \mathbf{x}^o as well. When the *missing mechanism* depends on missing values \mathbf{x}^m , the data are termed to be *missing not at random* (MNAR). From (3.10), an explicit form has to be assumed for the distribution of the missingness, and both the accuracy and the computational efficiency should be concerned.

When missingness is not totally controlled, as in most realistic applications, we cannot tell from the data alone whether the MCAR or MAR assumption is valid. Since the MCAR or MAR assumption is unlikely to be precisely satisfied in practice, inference based on these assumptions may lead to a bias. However, as demonstrated in many cases, it is believed that for realistic problems departures from MAR are

usually not large enough to significantly impact the analysis [CSK01]. On the other hand, without the MAR assumption, one must explicitly specify a model for the missingness \mathbf{R} , which is a difficult task in most cases. As a result, the data are typically assumed to be either MCAR or MAR in the literature, unless significant correlations between the missing values and the distribution of the missingness are suspected.

In this work we make the MAR assumption, and thus expression (3.11) applies. In the iQGME framework, the joint likelihood may be further expanded as

$$p(y, \mathbf{x}^o | \boldsymbol{\theta}) = \int p(y, \mathbf{x} | \boldsymbol{\theta}) d\mathbf{x}^m = \int_{ty>0} \int p(t | \mathbf{x}, \boldsymbol{\theta}_2) p(\mathbf{x} | \boldsymbol{\theta}_1) d\mathbf{x}^m dt. \quad (3.12)$$

The solution to such a problem with incomplete data \mathbf{x}^m is analytical since the distributions of t and \mathbf{x} are assumed to be a Gaussian and a Gaussian mixture model, respectively. Naturally, the missing features could be regarded as hidden variables to be inferred and the graphical representation of the iQGME with incomplete data remains the same as in Figure 3.1, except that the node presenting features are partially observed now. As elaborated below, the important but mild assumption that the features are distributed as a GMM enables us to analytically infer the variational distributions associated with the missing values in a procedure of variational Bayesian inference.

As in many models [W LX⁺07], estimating the distribution of the missing values first and learning the classifier at a second step gives the flexibility of selecting the classifier for the second step. However, (3.12) suggests that the classifier and the data distribution are coupled, provided that partial data are missing and thus have to be integrated out. Therefore, a joint estimation of missing features and classifiers (searching in the space of $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$) is more desirable than a two-step process (searching in the space of $\boldsymbol{\theta}_1$ for the distribution of the data, and then in the space of $\boldsymbol{\theta}_2$ for the classifier).

3.4 Variational Bayesian Inference

3.4.1 Basic construction

For simplicity we denote the collection of hidden variables and model parameters as Θ and specified hyper-parameters as Ψ . In a Bayesian framework we are interested in $p(\Theta|\mathcal{D}, \Psi)$, the joint posterior distribution of the unknowns given observed data and hyper-parameters. From Bayes' rule,

$$p(\Theta|\mathcal{D}, \Psi) = \frac{p(\mathcal{D}|\Theta)p(\Theta|\Psi)}{p(\mathcal{D}|\Psi)}, \quad (3.13)$$

where $p(\mathcal{D}|\Psi) = \int p(\mathcal{D}|\Theta)p(\Theta|\Psi)d\Theta$ is the marginal likelihood that often involves multi-dimensional integrals. Since these integrals are nonanalytical in most cases, the computation of the marginal likelihood is the principal challenge in Bayesian inference. These integrals are circumvented if only a point estimate $\hat{\Theta}$ is pursued, as in the expectation-maximization algorithm [DLR77]. Markov chain Monte Carlo (MCMC) sampling methods [GHRPS90, Nea93] provide one class of approximations for the full posterior, based on samples from a Markov chain whose stationary distribution is the posterior of interest. As a Markov chain is guaranteed to converge to its true posterior theoretically as long as the chain is long enough, MCMC samples constitute an unbiased estimation for the posterior. Most previous applications with a Dirichlet process prior [IJ01, WME94], including the related papers we reviewed in Section 3.1, have been implemented with various MCMC methods. The main concerns of MCMC methods are associated with computational costs for collecting sufficient samples and the difficulty of convergence diagnosis.

As an efficient alternative, the variational Bayesian (VB) method [Bea03] approximates the true posterior $p(\Theta|\mathcal{D}, \Psi)$ with a variational distribution $q(\Theta)$ with free variational parameters. The problem of computing the posterior is reformulated as an optimization problem of minimizing the Kullback-Leibler (KL) divergence be-

tween $q(\Theta)$ and $p(\Theta|\mathcal{D}, \Psi)$, which is equivalent to maximizing a lower bound of $\log p(\mathcal{D}|\Psi)$, the log marginal likelihood. This optimization problem can be solved iteratively with two assumptions on $q(\Theta)$: (i) $q(\Theta)$ is factorized; (ii) the factorized components of $q(\Theta)$ come from the same exponential family as the corresponding priors do. Since the lower bound cannot achieve the true log marginal likelihood in general, the approximation given by the variational Bayesian method is biased. Another issue concerning the VB algorithm is that the solution may be trapped at local optima since the optimization problem is not convex. The main advantages of the VB include the ease of convergence diagnosis and computational efficiency. As the VB is solving an optimization problem, the objective function – the lower bound of the log marginal likelihood – is a natural criterion for convergence diagnosis. Therefore, the VB is a good alternative to the MCMC especially when conjugacy is achieved and computational efficiency is desired. In recent publications [BJ06, KWT07], discussions on the implementation of the variational Bayesian inference are given for Dirichlet process mixtures.

We implement the variational Bayesian inference throughout this chapter, with comparisons made to Gibbs sampling. Since it is desirable to maintain the dependencies among random variables (*e.g.*, shown in the graphical models Figure 3.1) in the variational distribution $q(\Theta)$, one typically only breaks those dependencies that bring difficulty to computation. In the subsequent inference for the iQGME, we retain some dependencies as unbroken. Following [BJ06], we employ stick-breaking representations with a truncation level N as variational distributions to approximate the infinite-dimensional random measures G .

We detail the variational Bayesian inference for the case of incomplete data. The inference for the complete-data case is similar, except that all feature vectors are fully observed and thus the step of learning missing values is skipped. To avoid repetition, a thorough procedure for the complete-data case is not included, with

differences from the incomplete-data case indicated.

3.4.2 Variational distributions specification and updating

For the iQGME the unknowns are $\Theta = \{\mathbf{t}, \mathbf{x}^m, \mathbf{z}, \mathbf{V}, \alpha, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}\}$, with hyperparameters $\Psi = \{\mathbf{m}_0, u_0, \mathbf{B}_0, \nu_0, \tau_{10}, \tau_{20}, \gamma_0, a_0, b_0\}$. We specify the factorized variational distributions as

$$q(\mathbf{t}, \mathbf{x}^m, \mathbf{z}, \mathbf{V}, \alpha, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}) \\ = \prod_{i=1}^n [q_{t_i}(t_i) q_{\mathbf{x}_i^{m_i}, z_i}(\mathbf{x}_i^{m_i}, z_i)] \prod_{h=1}^{N-1} q_{V_h}(V_h) \prod_{h=1}^N [q_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h}(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) q_{\mathbf{w}_h}(\mathbf{w}_h)] \prod_{p=1}^{P+1} q_{\zeta_p, \lambda_p}(\zeta_p, \lambda_p) q_{\alpha}(\alpha)$$

where

- $q_{t_i}(t_i)$ is a truncated normal distribution,

$$t_i \sim \mathcal{TN}(\mu_i^t, 1, y_i t_i > 0), \quad i = 1, \dots, n,$$

which means the density function of t_i is assumed to be normal with mean μ_i^t and unit variance for those t_i satisfying $y_i t_i > 0$.

- $q_{\mathbf{x}_i^{m_i}, z_i}(\mathbf{x}_i^{m_i}, z_i) = q_{\mathbf{x}_i^{m_i}}(\mathbf{x}_i^{m_i} | z_i) q_{z_i}(z_i)$, where $q_{z_i}(z_i)$ is a multinomial distribution with probabilities $\boldsymbol{\rho}_i$, and there are N possible outcomes, $z_i \sim \mathcal{M}_N(1, \rho_{i1}, \dots, \rho_{iN})$, $i = 1, \dots, n$. Given the associated indicators z_i , since features are assumed to be distributed as a multivariate Gaussian, the distributions of missing values $\mathbf{x}_i^{m_i}$ are still Gaussian according to conditional properties of multivariate Gaussian distributions:

$$(\mathbf{x}_i^{m_i} | z_i = h) \sim \mathcal{N}_{|m_i|}(\mathbf{m}_h^{m_i | o_i}, \boldsymbol{\Sigma}_h^{m_i | o_i}), \quad i = 1, \dots, n, \quad h = 1, \dots, N.$$

We retain the dependency between $\mathbf{x}_i^{m_i}$ and z_i in the variational distribution since the inference is still tractable; for complete data, the variation distribution for $(\mathbf{x}_i^{m_i} | z_i = h)$ is not necessary.

- $q_{V_h}(V_h)$ is a beta distribution,

$$V_h \sim Be(v_{h1}, v_{h2}), \quad h = 1, \dots, N - 1.$$

Recall that we have a truncation level of N , which implies that the mixture proportions $\pi_h(\mathbf{V})$ are equal to zero for $h > N$. Therefore, $q_{V_h}(V_h) = \delta_1$ for $h = N$, and $q_{V_h}(V_h) = \delta_0$ for $h > N$. For $h < N$, V_h has a variational Beta posterior.

- $q_{\mu_h, \Lambda_h}(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h)$ is a normal-Wishart distribution,

$$(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) \sim \mathcal{N}_P(\mathbf{m}_h, u_h^{-1} \boldsymbol{\Lambda}_h^{-1}) \mathcal{W}(\mathbf{B}_h, \nu_h), \quad h = 1, \dots, N.$$

- $q_{\mathbf{w}_h}(\mathbf{w}_h)$ is a normal distribution,

$$\mathbf{w}_h \sim \mathcal{N}_{P+1}(\boldsymbol{\mu}_h^w, \boldsymbol{\Sigma}_h^w), \quad h = 1, \dots, N.$$

- $q_{\zeta_p, \lambda_p}(\zeta_p, \lambda_p)$ is a normal-gamma distribution,

$$(\zeta_p, \lambda_p) \sim \mathcal{N}(\phi_p, \gamma^{-1} \lambda_p^{-1}) Ga(a_p, b_p), \quad p = 1, \dots, P + 1.$$

- $q_\alpha(\alpha)$ is a Gamma distribution,

$$\alpha \sim Ga(\tau_1, \tau_2).$$

Given the specifications on the variational distributions, a mean-field variational algorithm [Bea03] is developed for the iQGME model. All update equations and derivations for $q(\mathbf{x}_i^{m_i}, z_i)$ are included. Each variational parameter is re-estimated iteratively conditioned on the current estimate of the others until the lower bound of the log marginal likelihood converges. Although the algorithm yields a bound for any initialization of the variational parameters, different initializations may lead to different bounds. To alleviate this local-maxima problem, one may perform multiple

independent runs with random initializations, and choose the run that produces the highest bound on the marginal likelihood. We will elaborate on our initializations in the experiment section.

For simplicity, we omit the subscripts on the variational distributions and henceforth use q to denote any variational distributions. In the following derivations and update equations, we use generic notation $\langle f \rangle_{q(\cdot)}$ to denote $\mathbb{E}_{q(\cdot)}[f]$, the expectation of a function f with respect to variational distributions $q(\cdot)$. The subscript $q(\cdot)$ is dropped when it shares the same arguments with f .

The update equations for cases with incomplete data are summarized as follows:

1. $q(t_i | \mu_i^t)$

$$\mu_i^t = \sum_{h=1}^N \rho_{ih} \langle \mathbf{w}_h \rangle^T \hat{\mathbf{x}}_{i,h}^b \quad \text{where } \hat{\mathbf{x}}_{i,h}^b = [\mathbf{x}_i^{o_i}; \mathbf{m}_h^{m_i|o_i}; 1].$$

The expectation of t_i and t_i^2 may be derived according to properties of truncated normal distributions:

$$\begin{aligned} \langle t_i \rangle &= \mu_i^t + \frac{\phi(-\mu_i^t)}{\mathbf{1}(y_i = 1) - \Phi(-\mu_i^t)}, \\ \langle t_i^2 \rangle &= 1 + (\mu_i^t)^2 + \frac{\mu_i^t \phi(-\mu_i^t)}{\mathbf{1}(y_i = 1) - \Phi(-\mu_i^t)}, \end{aligned}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the probability density function and the cumulative density function of the standard normal distribution, respectively.

2. $q(\mathbf{x}_i^{m_i}, z_i | \mathbf{m}_{ih}^{m_i|o_i}, \Sigma_{ih}^{m_i|o_i}, \rho_{ih})$

A related derivation for a GMM model with incomplete data could be found in [WLX⁺07], where no classifier terms appear.

First, we explicitly write the intercept w_h^b , that is, $\mathbf{w}_h = [(\mathbf{w}_h^x)^T, w_h^b]^T$:

$$\begin{aligned}
& q(\mathbf{x}_i^{m_i}, z_i = h) \\
\propto & \exp\{\langle \ln[p(t_i|z_i = h, \mathbf{x}_i, \mathbf{W})p(z_i = h|V)p(\mathbf{x}_i|z_i = h, \boldsymbol{\mu}, \boldsymbol{\Lambda})] \rangle_{q(t_i)q(\mathbf{w}_h)q(V)q(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h)}\} \\
\propto & A_{ih} \mathcal{N}_P(\mathbf{x}_i | \tilde{\boldsymbol{\mu}}_{ih}, \tilde{\boldsymbol{\Sigma}}_{ih}),
\end{aligned}$$

where

$$\begin{aligned}
\tilde{\boldsymbol{\Sigma}}_{ih} &= [\langle \mathbf{w}_h^x (\mathbf{w}_h^x)^T \rangle + \nu_h \mathbf{B}_h]^{-1} \\
\tilde{\boldsymbol{\mu}}_{ih} &= \tilde{\boldsymbol{\Sigma}}_{ih} [\langle t_i \rangle \langle \mathbf{w}_h^x \rangle + \nu_h \mathbf{B}_h \mathbf{m}_h - \langle \mathbf{w}_h^x w_h^b \rangle] \\
A_{ih} &= \exp\{\langle \ln V_h \rangle + \sum_{l < h} \langle \ln(1 - V_l) \rangle + \langle t_i \rangle \langle w_h^b \rangle \\
&\quad + \frac{1}{2} [\langle \ln |\boldsymbol{\Lambda}_h| \rangle + \tilde{\boldsymbol{\mu}}_{ih}^T \tilde{\boldsymbol{\Sigma}}_{ih}^{-1} \tilde{\boldsymbol{\mu}}_{ih} + \ln |\tilde{\boldsymbol{\Sigma}}_{ih}| - \frac{P}{u_h} - \mathbf{m}_h^T \nu_h \mathbf{B}_h \mathbf{m}_h - \langle (w_h^b)^2 \rangle]\}.
\end{aligned}$$

Since

$$\begin{bmatrix} \mathbf{x}_i^{o_i} \\ \mathbf{x}_i^{m_i} \end{bmatrix} \sim \mathcal{N}_P \left(\begin{bmatrix} \tilde{\boldsymbol{\mu}}_{ih}^{o_i} \\ \tilde{\boldsymbol{\mu}}_{ih}^{m_i} \end{bmatrix}, \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{ih}^{o_i o_i} & \tilde{\boldsymbol{\Sigma}}_{ih}^{o_i m_i} \\ \tilde{\boldsymbol{\Sigma}}_{ih}^{m_i o_i} & \tilde{\boldsymbol{\Sigma}}_{ih}^{m_i m_i} \end{bmatrix} \right),$$

the conditional distribution of missing features $\mathbf{x}_i^{m_i}$ given observable features $\mathbf{x}_i^{o_i}$ is also a normal distribution, that is, $\mathbf{x}_i^{m_i} | \mathbf{x}_i^{o_i} \sim \mathcal{N}_{|m_i|}(\mathbf{m}_h^{m_i|o_i}, \boldsymbol{\Sigma}_h^{m_i|o_i})$ with

$$\begin{aligned}
\mathbf{m}_h^{m_i|o_i} &= \tilde{\boldsymbol{\mu}}_{ih}^{m_i} + \tilde{\boldsymbol{\Sigma}}_{ih}^{m_i o_i} (\tilde{\boldsymbol{\Sigma}}_{ih}^{o_i o_i})^{-1} (\mathbf{x}_i^{o_i} - \tilde{\boldsymbol{\mu}}_{ih}^{o_i}), \\
\boldsymbol{\Sigma}_h^{m_i|o_i} &= \tilde{\boldsymbol{\Sigma}}_{ih}^{m_i m_i} - \tilde{\boldsymbol{\Sigma}}_{ih}^{m_i o_i} (\tilde{\boldsymbol{\Sigma}}_{ih}^{o_i o_i})^{-1} \tilde{\boldsymbol{\Sigma}}_{ih}^{o_i m_i}.
\end{aligned}$$

Therefore, $q(\mathbf{x}_i^{m_i}, z_i = h)$ could be factorized as the product of a factor independent of $\mathbf{x}_i^{m_i}$ and the variational posterior of $\mathbf{x}_i^{m_i}$, that is,

$$\begin{aligned}
q(\mathbf{x}_i^{m_i}, z_i = h) &\propto A_{ih} \mathcal{N}_{|o_i|}(\mathbf{x}_i^{o_i} | \tilde{\boldsymbol{\mu}}_{ih}^{o_i}, \tilde{\boldsymbol{\Sigma}}_{ih}^{o_i o_i}) \mathcal{N}_{|m_i|}(\mathbf{x}_i^{m_i} | \mathbf{m}_h^{m_i|o_i}, \boldsymbol{\Sigma}_h^{m_i|o_i}) \\
\rho_{ih} &\propto A_{ih} \mathcal{N}_{|o_i|}(\mathbf{x}_i^{o_i} | \tilde{\boldsymbol{\mu}}_{ih}^{o_i}, \tilde{\boldsymbol{\Sigma}}_{ih}^{o_i o_i})
\end{aligned}$$

For complete data, no factorization for the distribution for $\mathbf{x}_i^{m_i}$ is necessary:

$$\begin{aligned} \rho_{ih} \propto & \exp\{\langle t_i \rangle \langle \mathbf{w}_h \rangle^T \mathbf{x}_i - \frac{1}{2} \mathbf{x}_i^T \langle \mathbf{w}_h \mathbf{w}_h^T \rangle \mathbf{x}_i + \langle \ln V_h \rangle \\ & + \sum_{l < h} \langle \ln(1 - V_l) \rangle + \frac{1}{2} \langle \ln |\mathbf{\Lambda}_h| \rangle - \frac{1}{2} \langle (\mathbf{x}_i - \boldsymbol{\mu}_h)^T \mathbf{\Lambda}_h (\mathbf{x}_i - \boldsymbol{\mu}_h) \rangle\} \end{aligned}$$

3. $q(V_h | v_{h1}, v_{h2})$

Similar updating could be found in [BJ06], except that we put a prior belief on α here instead of setting a fixed number.

$$\begin{aligned} v_{h1} &= 1 + \sum_{i=1}^n \rho_{ih}, \quad v_{h2} = \langle \alpha \rangle + \sum_{i=1}^n \sum_{l > h} \rho_{il}; \\ \langle \ln V_h \rangle &= \psi(v_{h1}) - \psi(v_{h1} + v_{h2}), \quad \langle \ln(1 - V_l) \rangle = \psi(v_{l2}) - \psi(v_{l1} + v_{l2}). \end{aligned}$$

4. $q(\boldsymbol{\mu}_h, \mathbf{\Lambda}_h | \mathbf{m}_h, u_h, \mathbf{B}_h, \nu_h)$

Similar updating could be found in [WLX⁺07].

$$\begin{aligned} \nu_h &= \nu_0 + N_h, \quad u_h = u_0 + N_h, \quad \mathbf{m}_h = \frac{u_0 \mathbf{m}_0 + N_h \bar{\mathbf{x}}_h}{u_h}, \\ \mathbf{B}_h^{-1} &= \mathbf{B}_0^{-1} + \sum_{i=1}^n \rho_{ih} \hat{\boldsymbol{\Omega}}_{i,h} + N_h \bar{\mathbf{S}}_h + \frac{u_0 N_h}{u_h} (\bar{\mathbf{x}}_h - \mathbf{m}_0)(\bar{\mathbf{x}}_h - \mathbf{m}_0)^T, \end{aligned}$$

where

$$\begin{aligned} N_h &= \sum_{i=1}^n \rho_{ih}, \quad \bar{\mathbf{x}}_h = \sum_{i=1}^n \rho_{ih} \mathbf{x}_i / N_h, \quad \bar{\mathbf{S}}_h = \sum_{i=1}^n \rho_{ih} (\hat{\mathbf{x}}_{i,h} - \bar{\mathbf{x}}_h)(\hat{\mathbf{x}}_{i,h} - \bar{\mathbf{x}}_h)^T / N_h. \\ \hat{\mathbf{x}}_{i,h} &= \begin{bmatrix} \mathbf{x}_i^{o_i} \\ \mathbf{m}_h^{m_i | o_i} \end{bmatrix}, \quad \hat{\boldsymbol{\Omega}}_{i,h} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_h^{m_i | o_i} \end{bmatrix}. \end{aligned}$$

$$\langle \ln |\mathbf{\Lambda}_h| \rangle = \sum_{p=1}^P \psi((\nu_h - p + 1)/2) + P \ln 2 + \ln |\mathbf{B}_h|,$$

$$\langle (\mathbf{x}_i - \boldsymbol{\mu}_h)^T \mathbf{\Lambda}_h (\mathbf{x}_i - \boldsymbol{\mu}_h) \rangle = (\hat{\mathbf{x}}_{i,h} - \mathbf{m}_h)^T \nu_h \mathbf{B}_h (\hat{\mathbf{x}}_{i,h} - \mathbf{m}_h) + P/u_h + \text{tr}(\nu_h \mathbf{B}_h \hat{\boldsymbol{\Omega}}_{i,h}).$$

$$5. q(\mathbf{w}_h | \boldsymbol{\mu}_h^w, \boldsymbol{\Sigma}_h^w), \langle \mathbf{w}_h \rangle = \boldsymbol{\mu}_h^w, \quad \langle \mathbf{w}_h \mathbf{w}_h^T \rangle = \boldsymbol{\Sigma}_h^w + \boldsymbol{\mu}_h^w (\boldsymbol{\mu}_h^w)^T.$$

$$\boldsymbol{\Sigma}_h^w = \left(\sum_{i=1}^n \rho_{ih} (\hat{\boldsymbol{\Omega}}_{i,h} + \hat{\mathbf{x}}_{i,h}^b \hat{\mathbf{x}}_{i,h}^{bT}) + \text{diag}(\langle \boldsymbol{\lambda} \rangle) \right)^{-1},$$

$$\boldsymbol{\mu}_h^w = \boldsymbol{\Sigma}_h^w \left(\sum_{i=1}^n \rho_{ih} \hat{\mathbf{x}}_{i,h}^b \langle t_i \rangle + \text{diag}(\langle \boldsymbol{\lambda} \rangle) \boldsymbol{\phi} \right).$$

$$6. q(\zeta_p, \lambda_p | \phi_p, \gamma, a_p, b_p), \langle \lambda_p \rangle = a_p / b_p.$$

Similar updating could be found in [XLCK07].

$$\phi_p = \sum_{h=1}^N \langle w_{hp} \rangle / \gamma, \quad \gamma = \gamma_0 + N$$

$$a_p = a_0 + \frac{N}{2}, \quad b_p = b_0 + \frac{1}{2} \sum_{h=1}^N \langle w_{hp}^2 \rangle - \frac{1}{2} \gamma \phi_p^2.$$

$$7. q(\alpha | \tau_1, \tau_2), \langle \alpha \rangle = \tau_1 / \tau_2.$$

Similar updating could be found in any VB-inferred DP model with a Gamma prior on α [XLCK07].

$$\tau_1 = N - 1 + \tau_{10}, \quad \tau_2 = \tau_{20} - \sum_{h=1}^{N-1} \langle \ln(1 - V_h) \rangle.$$

3.4.3 Prediction

For a new observed feature vector $\mathbf{x}_\star^{o_\star}$, the prediction on the associated class label y_\star is given by integrating out the missing values.

$$\begin{aligned} P(y_\star = 1 | \mathbf{x}_\star^{o_\star}, \mathcal{D}) &= \frac{p(y_\star = 1, \mathbf{x}_\star^{o_\star} | \mathcal{D})}{p(\mathbf{x}_\star^{o_\star} | \mathcal{D})} = \frac{\int p(y_\star = 1, \mathbf{x}_\star | \mathcal{D}) d\mathbf{x}_\star^{m_\star}}{\int p(\mathbf{x}_\star | \mathcal{D}) d\mathbf{x}_\star^{m_\star}} \\ &= \frac{\int \sum_{h=1}^N P(z_\star = h | \mathcal{D}) p(x_\star | z_\star = h, \mathcal{D}) P(y_\star = 1 | \mathbf{x}_\star, z_\star = h, \mathcal{D}) d\mathbf{x}_\star^{m_\star}}{\int \sum_{k=1}^N P(z_\star = k | \mathcal{D}) p(x_\star | z_\star = k, \mathcal{D}) d\mathbf{x}_\star^{m_\star}} \end{aligned}$$

We marginalize the hidden variables over their variational distributions to compute the predictive probability of the class label

$$\begin{aligned}
& P(y_\star = 1 | \mathbf{x}_\star^{o_\star}, \mathcal{D}) \\
&= \frac{\sum_{h=1}^N \mathbb{E}_V[\pi_h] \int_0^\infty \int \mathbb{E}_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h}[\mathcal{N}_P(\mathbf{x}_\star | \boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})] \mathbb{E}_{\mathbf{w}_h}[\mathcal{N}(t_\star | \mathbf{w}_h^T \mathbf{x}_\star^b, 1)] d\mathbf{x}_\star^{m_\star} dt_\star}{\sum_{k=1}^N \mathbb{E}_V[\pi_k] \int \mathbb{E}_{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k}[\mathcal{N}_P(\mathbf{x}_\star | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})] d\mathbf{x}_\star^{m_\star}}
\end{aligned}$$

where

$$\mathbb{E}_V[\pi_h] = \mathbb{E}_V[V_h \prod_{l < h} (1 - V_l)] = \langle V_h \rangle \prod_{l < h} \langle 1 - V_l \rangle = \left[\frac{v_{h1}}{v_{h1} + v_{h2}} \right]^{\mathbf{1}(h < N)} \prod_{l < h} \left[\frac{v_{l2}}{v_{l1} + v_{l2}} \right]^{\mathbf{1}(h > 1)}$$

The expectation $\mathbb{E}_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h}[\mathcal{N}_P(\mathbf{x}_\star | \boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})]$ is a multivariate Student-t distribution [Att00]. However, for the incomplete-data situation, the integral over the missing values is tractable only when the two terms in the integral are both normal. To retain the form of norm distributions, we use the posterior means of $\boldsymbol{\mu}_h$, $\boldsymbol{\Lambda}_h$ and \mathbf{w}_h to approximate the variables:

$$\begin{aligned}
& P(y_\star = 1 | \mathbf{x}_\star^{o_\star}, \mathcal{D}) \\
&\approx \frac{\sum_{h=1}^N \mathbb{E}_V[\pi_h] \int_0^\infty \int \mathcal{N}_P(\mathbf{x}_\star | \mathbf{m}_h, (\nu_h \mathbf{B}_h)^{-1}) \mathcal{N}(t_\star | (\boldsymbol{\mu}_h^w)^T \mathbf{x}_\star^b, 1) d\mathbf{x}_\star^{m_\star} dt_\star}{\sum_{k=1}^N \mathbb{E}_V[\pi_k] \int \mathcal{N}_P(\mathbf{x}_\star | \mathbf{m}_k, (\nu_k \mathbf{B}_k)^{-1}) d\mathbf{x}_\star^{m_\star}} \\
&= \frac{\sum_{h=1}^N \mathbb{E}_V[\pi_h] \mathcal{N}_{|o_\star|}(\mathbf{x}_\star^{o_\star} | \mathbf{m}_h^{o_\star}, (\nu_h \mathbf{B}_h)^{-1, o_\star o_\star}) \int_0^\infty \mathcal{N}(t_\star | \varphi_{\star h}, g_{\star h}) dt_\star}{\sum_{k=1}^N \mathbb{E}_V[\pi_k] \mathcal{N}_{|o_\star|}(\mathbf{x}_\star^{o_\star} | \mathbf{m}_k^{o_\star}, (\nu_k \mathbf{B}_k)^{-1, o_\star o_\star})}
\end{aligned}$$

where

$$\begin{aligned}
\varphi_{\star h} &= [\mathbf{m}_h^T, 1] \boldsymbol{\mu}_h^w + \boldsymbol{\Gamma}_{\star h}^T (\boldsymbol{\Delta}_h^{o_\star o_\star})^{-1} (\mathbf{x}_\star^{o_\star} - \mathbf{m}_h^{o_\star}) \\
g_{\star h} &= 1 + (\bar{\boldsymbol{\mu}}_h^w)^T \boldsymbol{\Delta}_h \bar{\boldsymbol{\mu}}_h^w - \boldsymbol{\Gamma}_{\star h}^T (\boldsymbol{\Delta}_h^{o_\star o_\star})^{-1} \boldsymbol{\Gamma}_{\star h} \\
\boldsymbol{\Gamma}_{\star h} &= \boldsymbol{\Delta}_h^{o_\star o_\star} (\boldsymbol{\mu}_h^w)^{o_\star} + \boldsymbol{\Delta}_h^{o_\star m_\star} (\boldsymbol{\mu}_h^w)^{m_\star} \\
\bar{\boldsymbol{\mu}}_h^w &= (\boldsymbol{\mu}_h^w)_{1:P} \\
\boldsymbol{\Delta}_h &= (\nu_h \mathbf{B}_h)^{-1}
\end{aligned}$$

For complete data the integral of missing features is absent, so we take advantage of the full variational posteriors for prediction.

3.4.4 Computational complexity

Given the data dimensionality P , the number of data points n , and the truncation level (or the number of clusters) N specified for the variational distributions, we compare the iQGME to closely related DP regression models [MO06, SN09], in terms of the time and memory complexity. The inference of the iQGME with complete data requires inversion of two $P \times P$ matrices (the covariance matrices for the inputs and the local expert) associated with each cluster. Therefore, the time and memory complexity are $O(2NP^3)$ and $O(2NP^2)$, respectively. With incomplete data, since the missing pattern is unique for each data point, the time and memory complexity increase with number of data points, *i.e.*, $O(nNP^3)$ and $O(nNP^2)$, respectively. The mixture of Gaussian process experts [MO06] requires $O(NP^3 + n^3/N)$ computations for each MCMC iteration if the N experts equally divide the data, and the memory complexity is $O(NP^2 + n^2/N)$. In the model proposed by [SN09], no matrix inversion is needed since the covariates are assumed to be independent. The time and memory complexity are $O(NP)$ and $O(NP)$, respectively.

From the aspect of computational complexity, the model in [MO06] is restricted by the increase of both dimensionality and data size; while the model proposed in [SN09] is more efficient. Although the proposed model requires more computations for each MCMC iteration than the latter one, we are able to handle missing values naturally, and much more efficiently compared to the former one. Considering the usual number of iterations required by the VB (several dozens) and the MCMC (thousands or even tens of thousands), our model is even more efficient.

3.5 Experimental Results

In all the following experiments the hyper-parameters are set as follows: $a_0 = 0.01$, $b_0 = 0.01$, $\gamma_0 = 0.1$, $\tau_{10} = 0.05$, $\tau_{20} = 0.05$, $u_0 = 0.1$, $\nu_0 = P + 2$, and \mathbf{m}_0 and \mathbf{B}_0 are set according to sample mean and sample precision, respectively. These parameters have not been optimized for any particular data set (which are all different in form), and the results are relatively insensitive to “reasonable” settings. The truncation level for the variational distribution is set to be $N = 20$. We have found the results insensitive to the truncation level, for values larger than that considered here.

Because of the local-maxima issue associated with the VB, initialization of the variational parameters is often important. We initialize most variational hyper-parameters using the corresponding prior hyper-parameters, which are data-independent. The precision/covariance matrices \mathbf{B}_h and Σ_h^w are simply initialized as identity matrices. However, for several other hyper-parameters, we may obtain information for good start points from the data. Specifically, the variational mean of the soft label μ_i^t is initialized by the associated label y_i . A K -means clustering algorithm is implemented on the feature vectors, and the cluster means and identifications for objects are used to initialize the variational mean of the Gaussian means \mathbf{m}_h and the indicator probabilities ρ_i , respectively. As an alternative, one may randomly initialize \mathbf{m}_h and ρ_i multiple times, and select the solution that produces the highest lower bound on the log marginal likelihood. The two approaches work almost equivalently for low-dimensional problems; however, for problems with moderate to high dimensionality, it could be fairly difficult to get a satisfying initialization by making several random trials.

3.5.1 Synthetic data

We first demonstrate the proposed iQGME model on a synthetic data set, for illustrative purposes. The data are generated according to a GMM model $p(\mathbf{x}) = \sum_{k=1}^3 \pi_k \mathcal{N}_2(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with the following parameters:

$$\boldsymbol{\pi} = [1/3 \quad 1/3 \quad 1/3], \quad \boldsymbol{\mu}_1 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.52 & -0.36 \\ -0.36 & 0.73 \end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.47 & 0.19 \\ 0.19 & 0.7 \end{bmatrix}, \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.52 & -0.36 \\ -0.36 & 0.73 \end{bmatrix}.$$

The class boundary for each Gaussian component is given by three lines $x_2 = w_k x_1 + b_k$ for $k = 1, 2, 3$, where $w_1 = 0.75, b_1 = 2.25$, $w_2 = -0.58, b_2 = 0.58$, and $w_3 = 0.75, b_3 = -3.75$. The simulated data are shown in Figure 3.2(a), where black dots and dashed ellipses represent the true means and covariance matrices of the Gaussian components, respectively.

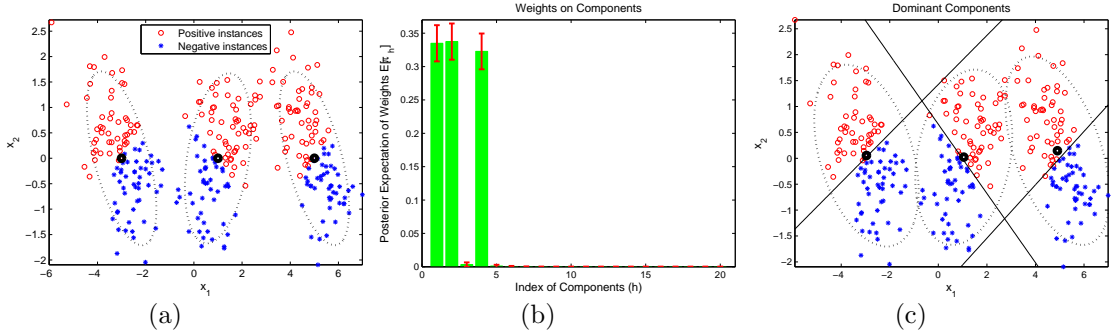


FIGURE 3.2: Synthetic three-Gaussian single-task data with inferred components. (a) Data in feature space with true labels and true Gaussian components indicated; (b) inferred posterior expectation of weights on components, with standard deviations depicted as error bars; (c) ground truth with posterior means of dominant components indicated (the linear classifiers and Gaussian ellipses are inferred from the data).

The inferred mean mixture weights with standard deviations are depicted in Figure 3.2(b), and it is observed that three dominant mixture components (local “experts”) are inferred. The dominant components (those with mean weight larger

than 0.005) are characterized by Gaussian means, covariance matrices and local experts, as depicted in Figure 3.2(c). From Figure 3.2(c), the nonlinear classification is manifested by using three dominant *local* linear classifiers, with a GMM defining the effective regions stochastically.

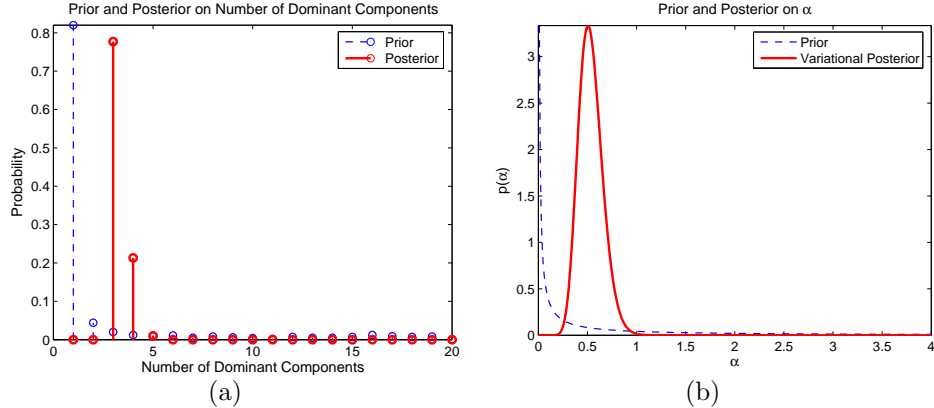


FIGURE 3.3: Synthetic three-Gaussian single-task data: (a) prior and posterior beliefs on the number of dominant components; (b) prior and posterior beliefs on α .

An important point is that we are not selecting a “correct” number of mixture components as in most mixture-of-expert models, including the finite QGME model [LLC07a]. Instead, there exists uncertainty on the number of components in our posterior belief. Since this uncertainty is not inferred directly, we obtain samples for the number of dominant components by calculating π_h based on V_h sampled from their probability density functions (prior or variational posterior), and the probability mass functions given by histogram are shown in Figure 3.3(a). As discussed, the scale parameter α is highly related to the number of clusters, so we depict the prior and the variational posterior on α in Figure 3.3(b).

The predictions in feature space are presented in Figure 3.4, where the prediction in sub-figure (a) is given by integrating over the full posteriors of local experts and parameters (means and covariance matrices) of Gaussian components; while the prediction in sub-figure (b) is given by posterior means. We examine these two cases

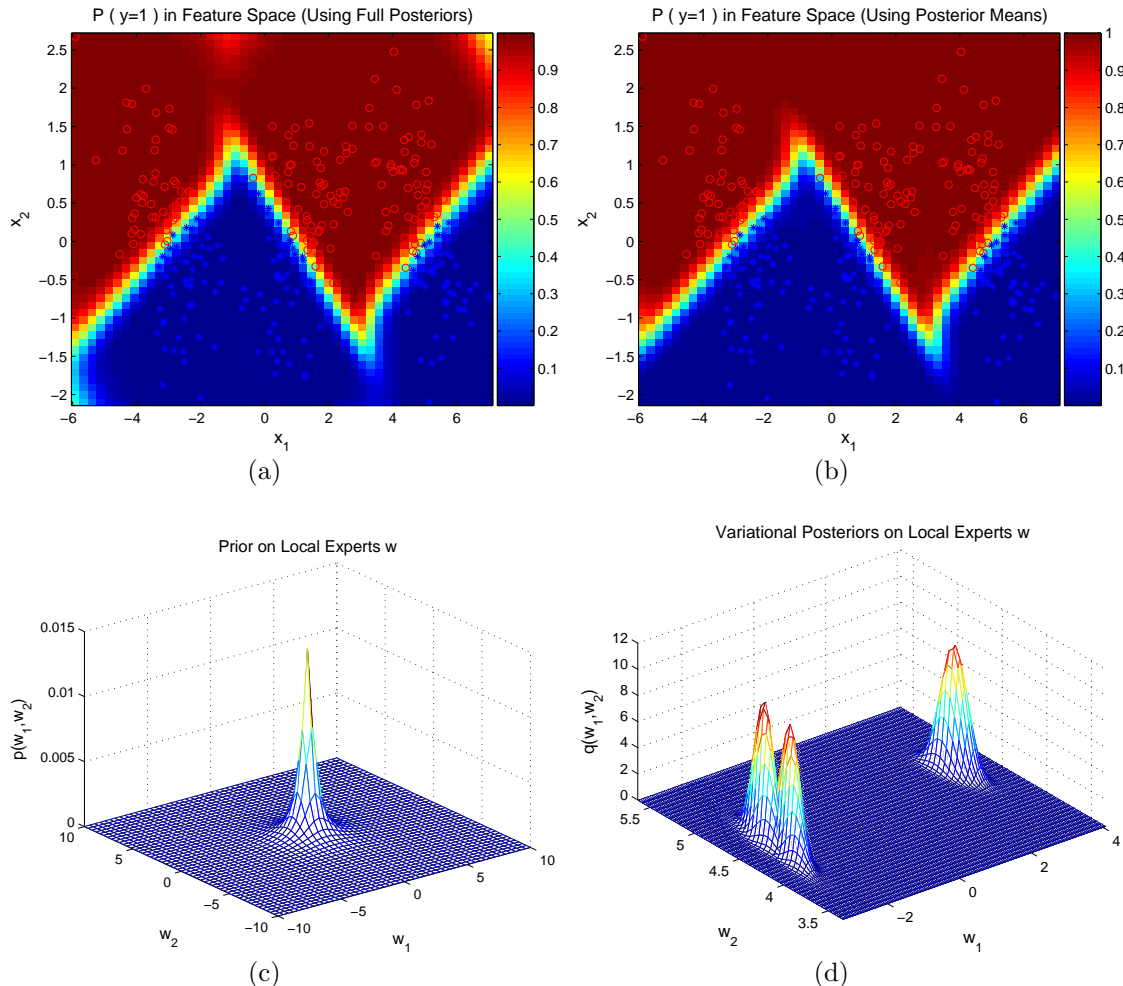


FIGURE 3.4: Synthetic three-Gauss single-task data: (a) prediction in feature space using the full posteriors; (b) prediction in feature space using the posterior means; (c) a common broad prior on local experts; (d) variational posteriors on local experts.

since the analytical integrals over the full posteriors may be unavailable sometimes in practice (for example, for cases with incomplete data as discussed in Section 3.4). From Figures 3.4(a) and 3.4(b), we observe that these two predictions are fairly similar, except that (a) allows more uncertainty on regions with scarce data. The reason for this is that the posteriors are often peaked and thus posterior means are usually representative. As an example, we plot the broad common prior imposed for local experts in Figure 3.4(c) and the peaked variational posteriors for three dominant

experts in Figure 3.4(d). According to Figure 3.4, we suggest the usage of full posteriors for prediction whenever integrals are analytical, that is, for experiments with complete data. It also empirically justifies the use of posterior means as an approximation. These results have been computed using VB inference, with MCMC-based results presented below, as a comparison.

3.5.2 Benchmark data

To further evaluate the proposed iQGME, we compare it with other models, using benchmark data sets available from the UCI machine learning repository [NHBM98]. Specifically, we consider Wisconsin Diagnostic Breast Cancer (WDBC) and the Johns Hopkins University Ionosphere database (Ionosphere) data sets, which have been studied in the literature [WLX⁺07, LLC07a]. These two data sets are summarized in Table 3.1.

Table 3.1: Details of Ionosphere and WDBC data sets.

| Data set | Dimension | Number of positive instances | Number of negative instances |
|------------|-----------|------------------------------|------------------------------|
| Ionosphere | 34 | 126 | 225 |
| WDBC | 30 | 212 | 357 |

The models we compare to include:

- State-of-the-art classification algorithms: Support Vector Machines (SVM) [Vap95] and Relevance Vector Machines (RVM) [Tip00]. We consider both linear models (Linear) and non-linear models with radial basis function (RBF) for both algorithms. For each data set, the kernel parameter is optimized given one training/test/validation separation, and then fixed for all the other experimental settings. The RVM models are implemented with Tipping’s Matlab code available at <http://www.miketipping.com/index.php?page=rvm>.

Since those SVM and RVM algorithms are not directly applicable to problems with missing features, we use two methods to impute the missing values before

the implementation. One is using the mean of observed values (unconditional mean) for the given feature, referred to as Uncond; the other is using the posterior mean conditional on observed features (conditional mean), referred to as Cond [SG02].

- Classifiers handling missing values: the finite QGME inferred by expectation-maximization (EM) [LLC07a], referred to as QGME-EM, and a two-stage algorithm [WLX⁺07] where the parameters of the GMM for the covariates are estimated first given the observed features, and then a marginalized linear logistic regression (LR) classifier is learned, referred to as LR-Integration. Results are cited from [LLC07a] and [WLX⁺07], respectively.

In order to simulate the *missing at random* setting, we randomly remove a fraction of feature values according to a uniform distribution, and assume the rest are observed. Any instance with all feature values missing is deleted. After that, we randomly split each data set into training and test subsets, imposing that each subset encompasses at least one instance from each of the classes. Note that the random pattern of missing features and the random partition of training and test subsets are independent of each other. By performing multiple trials we consider the general (average) performance for various data settings. For convenient comparison with [WLX⁺07] and [LLC07a], the performance of algorithms is evaluated in terms of the area under a receiver operating characteristic (ROC) curve (AUC) [HM82].

The results on the Ionosphere and WDBC data sets are summarized in Figures 3.5 and 3.6, respectively, where we consider 25% and 50% of the feature values missing. Given a portion of missing values, each curve is a function of the fraction of data used in training. For a given size of training data, we perform ten independent trials for the SVM and RVM models and the proposed iQGME.

From both Figures 3.5 and 3.6, the proposed iQGME-VB is robust for all the

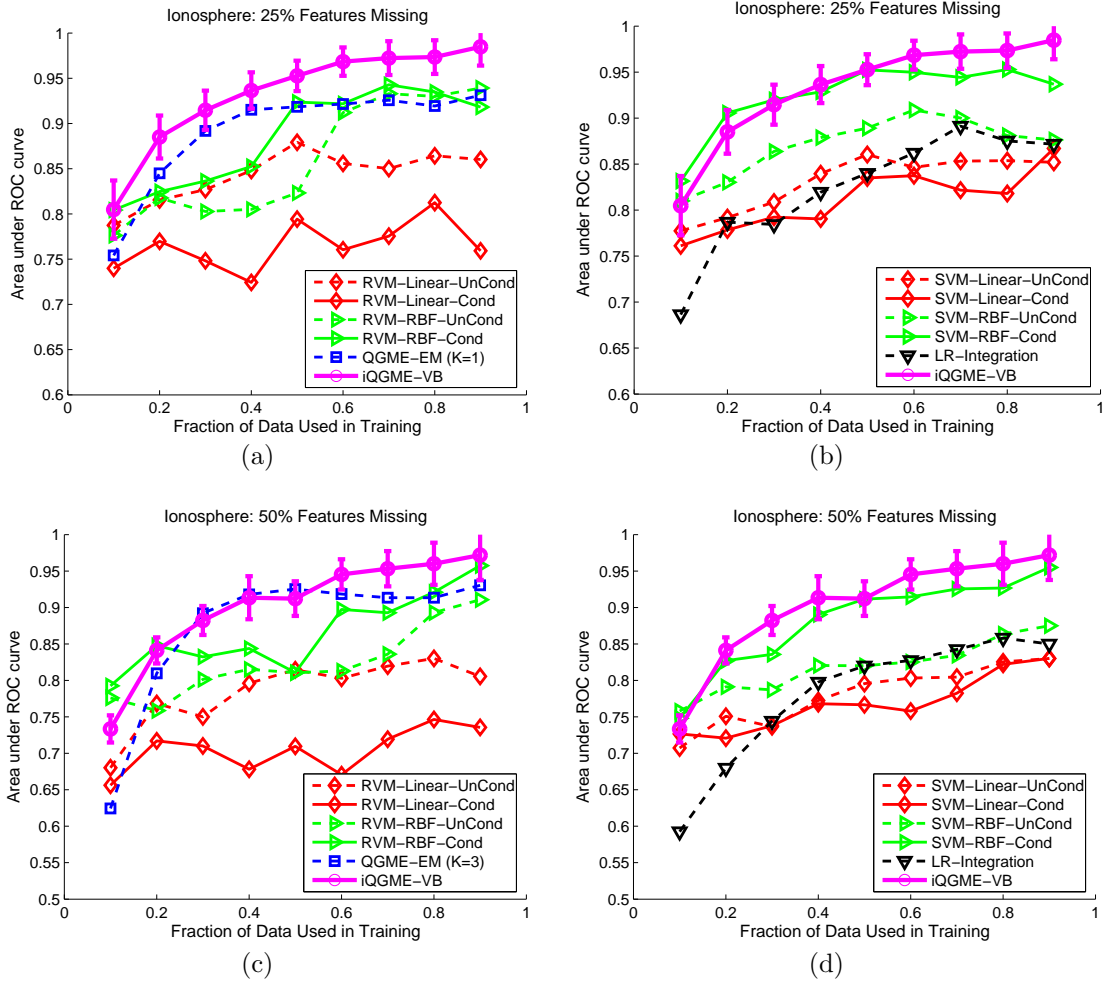


FIGURE 3.5: Results on Ionosphere data set for (a)(b) 25%, and (c)(d) 50% of the feature values missing. For legibility, we only report the standard deviation for the proposed iQGME-VB algorithm as error bars, and present the compared algorithms in two figures for each case. The results of the finite QGME solved with an expectation-maximization method are cited from [LLC07a], and those of LR-Integration are cited from [WLX⁺07]. Since the performance of the QGME-EM is affected by the choice of number of experts K , the overall best results among $K = 1, 3, 5, 10, 20$ are cited for comparison in each case (no such selection of K is required for the proposed iQGME-VB algorithm).

experimental settings, and its overall performance is the best among all algorithms considered. Although the RVM-RBF-Cond and the SVM-RBF-Cond perform well for the Ionosphere data set, especially when the training data is limited, their performance on the WDBC data set is not as good. The kernel methods benefit from the introduction of the RBF kernel for the Ionosphere data set; however, the performance

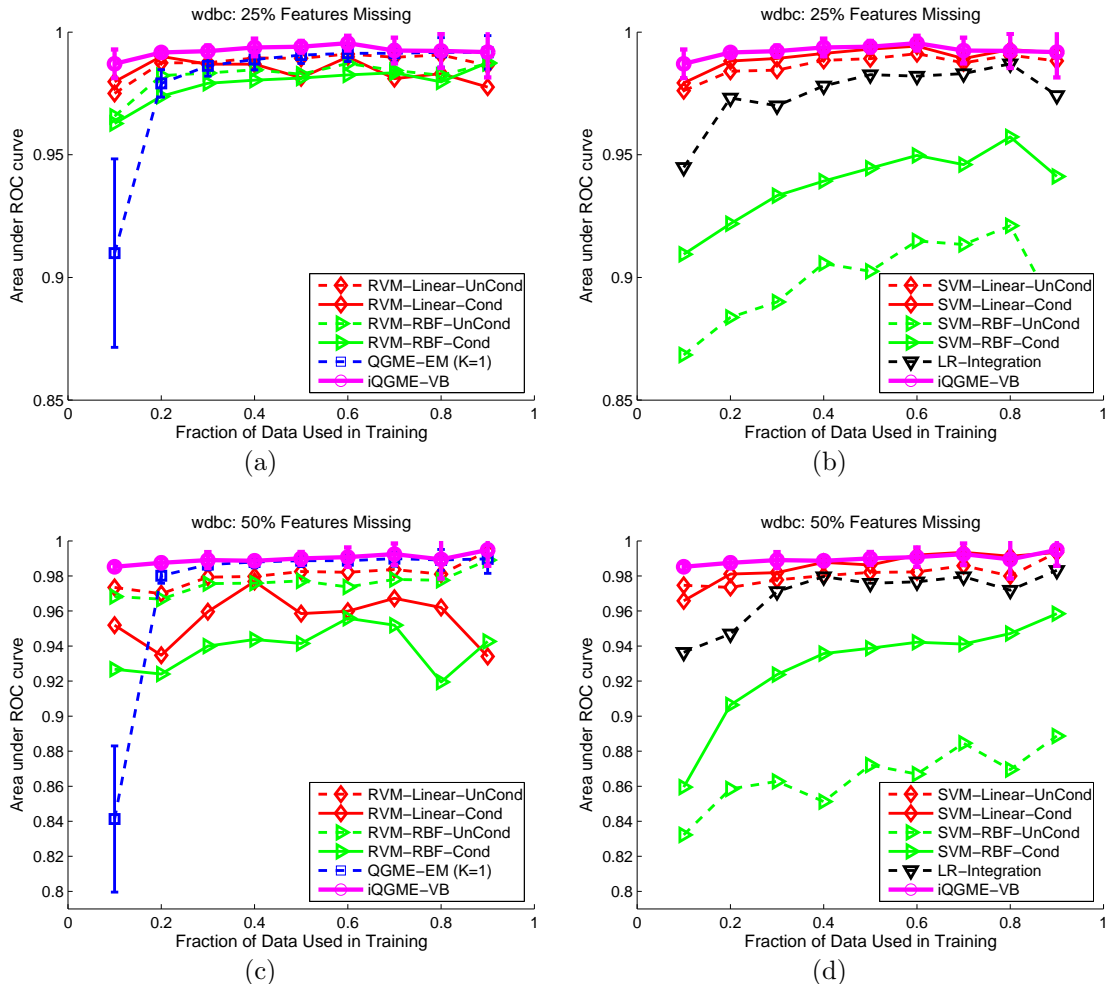


FIGURE 3.6: Results on WDBC data set for cases when (a)(b) 25%, and (c)(d) 50% of the feature values are missing. Refer to Figure 3.5 for additional information.

is inferior for the WDBC data set. We also note that the one-step iQGME and the finite QGME outperform the two-step LR-integration. The proposed iQGME consistently performs better than the finite QGME (where, for the latter, in all cases we show results for the best/optimized choice of number of experts K), which reveals the advantage of retaining the uncertainty on the model structure (number of experts) and model parameters. As shown in Figure 3.6, the advantage of considering the uncertainty on the model parameters is fairly pronounced for the WDBC data set, especially when training examples are relatively scarce and thus the point-

estimation EM method suffers from over-fitting issues. A more detailed examination on the model uncertainty is shown in Figures 3.7 and 3.8.

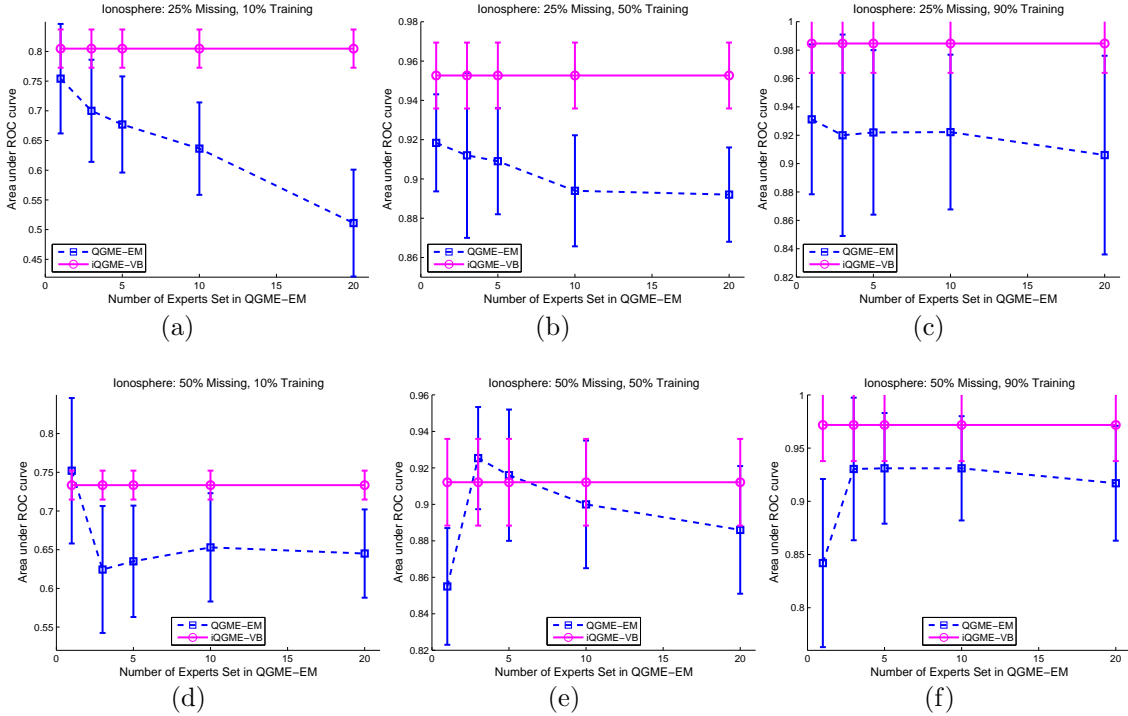


FIGURE 3.7: The comparison on the Ionosphere data set between QGME-EM with different preset number of clusters K and the proposed iQGME-VB, when (a)(b)(c) 25%, and (d)(e)(f) 50% of the features are missing. In each row, 10%, 50%, and 90% of samples are used for training, respectively. Results of QGME-EM are cited from [LLC07a].

In Figure 3.7, the influence of the preset value for K on the QGME-EM model is examined on the Ionosphere data. We observe that with different fractions of missing values and training samples, the values for K which achieve the best performance may be different; as K goes to a large number (e.g., 20 here), the performance gets worse due to over-fitting. In contrast, we do not need to set the number of clusters for the proposed iQGME-VB model. As long as the truncation level N is large enough ($N = 20$ for all the experiments), the number of clusters is inferred by the algorithm. We give an example for the posterior on the number of clusters inferred by the proposed iQGME-VB model, and report the statistics for the most probable

number of experts given each missing fraction and training fraction in Figure 3.8, which suggests that the number of clusters may vary significantly even for the trials with the same fraction of feature values missing and the same fraction of samples for training. Therefore, it may be not appropriate to set a fixed value for the number of clusters for all the experimental settings as one has to do for the QGME-EM.

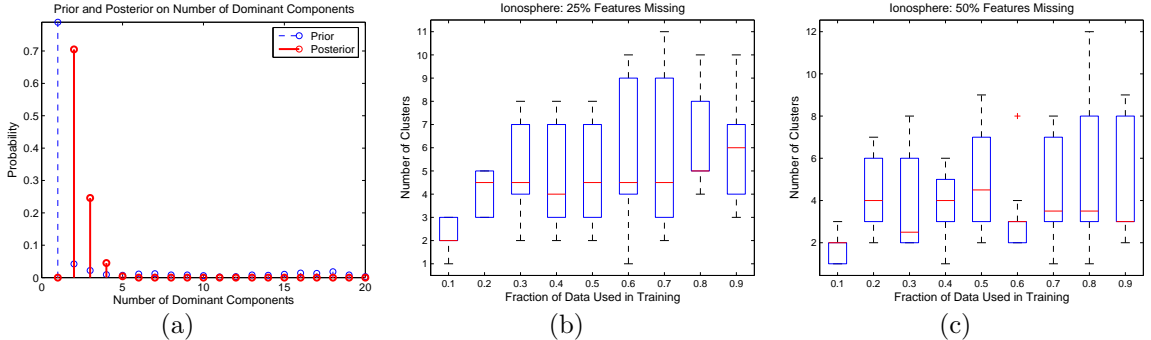


FIGURE 3.8: Number of clusters for the Ionosphere data set inferred by iQGME-VB. (a) Prior and inferred posterior on the number of clusters for one trial given 10% samples for training. The number of clusters for the case when (b) 25%, and (c) 50% of features are missing. The most probable value of clusters number is used for each trial to generate (b) and (c) (e.g, the most probable value of clusters number is two for the trial shown in (a)). In (b) and (c), the distribution of number of clusters for the ten trials given each missing fraction and training fraction is presented as a box-plot, where the red line represents the median; the bottom and top of the blue box are the 25th and 75th percentile, respectively; the bottom and top black lines are the end of the whiskers, which could be the minimum and maximum, respectively; if some data are beyond 1.5 times of the length of the blue box (interquartile range), they are outliers, indicated by a red ‘+’.

Although our main purpose is classification, one may also be interested in how well the algorithm can estimate the missing values while pursuing the main purpose. In Figure 3.9, we show the ratio of correctly estimated missing values for the Ionosphere data set with 25% feature values missing, where two criteria are considered: true values are less than one standard deviation (red circles) or two standard deviations (blue squares) away from the posterior means. This figure suggests that the algorithm estimates most of the missing values in a reasonable range away from the true values when the training size is large enough; even with not so satisfying estimations (as for

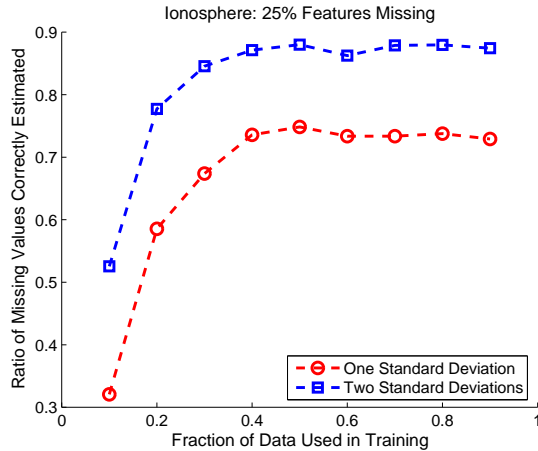


FIGURE 3.9: Ratio of missing values whose true values are less than one standard deviation (red circles) or two standard deviations (blue squares) away from the posterior means for the Ionosphere data set with 25% feature values missing. One trial for each training size is considered.

limited training data), the classification results are still relatively robust as shown in Figure 3.5.

We have discussed the advantages and disadvantages for the inference with MCMC and VB in Section 3.4.1. Here we take the Ionosphere data with 25% features missing as an example to compare these two inference techniques, as shown in Figure 3.10. It can be seen that they achieve similar performance for the particular iQGME model proposed in this paper. The time consumed for each iteration is also comparable, and increases almost linearly with the training size, as discussed in Section 3.4.4. The VB inference takes a little bit longer per iteration, probably due to the extra computation for the lower bound of the log marginal likelihood, which serves as convergence criterion. Significant differences occur on the number of iterations we have to take. In the experiment, even though we set a very strict threshold (10^{-6}) for the relative change of the lower bound, the VB algorithm converges at about 50 iterations for most cases except when training data are very scarce (10%). For the MCMC inference, we discard the initial samples from the first 1000 iterations (burn-in), and collect the next 500 samples to present the posterior. It is far from enough

to claim convergence; however, we consider it a fair comparison for computation as the two methods yield similar results under this setting. Given the fact that the VB algorithm only takes about 1/30 the CPU time, and VB and MCMC performance are similar, in the following examples we only present results based on VB inference. However, in all the examples below we also performed Gibbs sampling, and the relative inference consistency and computational costs relative to VB were found to be as summarized here (*i.e.*, in all cases there was close agreement between the VB and MCMC inferences, and considerable computational acceleration manifested by VB).

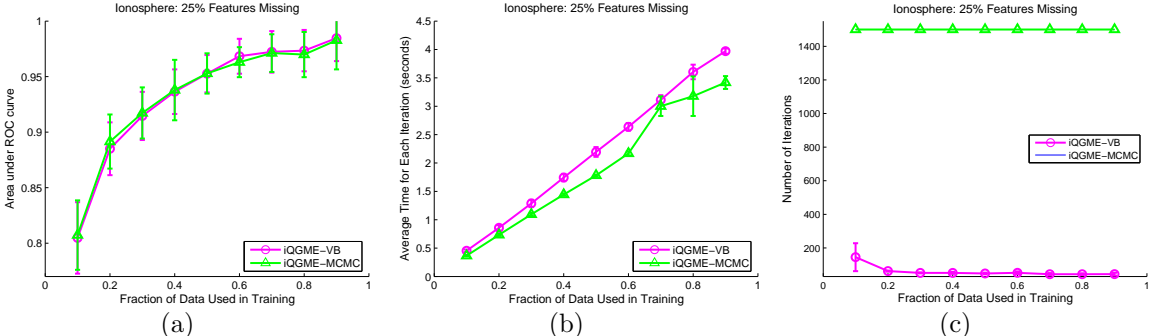


FIGURE 3.10: Comparison between VB and MCMC inferred iQGME on the Ionosphere data with 25% features missing in terms of (a) performance, (b) time consumed for each iteration, and (c) number of iterations. For the VB inference, we set a threshold (10^{-6}) for the relative change of lower bound in two consecutive iterations as the convergence criterion; for the MCMC inference, we discard the initial samples from the first 1000 iterations (burn-in), and collect the next 500 samples to present the posterior.

3.5.3 Unexploded ordnance data

We now consider an unexploded ordnance (UXO) detection problem [ZCY⁺03], where two types of sensors are used to collect data, but one of them may be absent for particular targets. Specifically, one sensor is a magnetometer (MAG) and the other an electromagnetic induction (EMI) sensor; these sensors are deployed separately to interrogate buried targets, and for some targets both sensors are deployed and for others only one sensor is deployed. This is a real sensing problem for which

missing data occurs naturally. The total number of targets are 146, where 79 of them are UXO and the rest are non-UXO (*i.e.*, non-explosives). A six-dimensional feature vector is extracted from the raw signals to represent each target, with the first three components corresponding to MAG features and the rest as EMI features (details on feature extraction is provided in [ZCY⁺03]). Figure 3.11 shows the missing patterns for this data set.

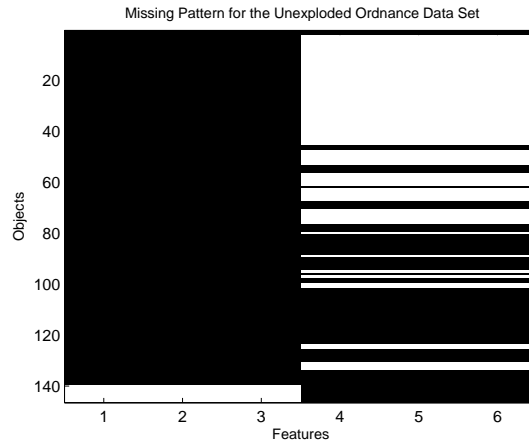


FIGURE 3.11: Missing pattern for the unexploded ordnance data set, where black and white indicate observed and missing, respectively.

We compare the proposed iQGME-VB algorithm with the SVM, RVM and LR-Integration as detailed in Section 3.5.2. In order to evaluate the overall performance of classifiers, we randomly partition the training and test subsets, and change the training size. The area under the ROC curve and the classification accuracy (using $P(y = 1) = 0.5$ as threshold) are two criteria considered. Results are shown in Figure 3.12, where only performance means are reported for the legibility of the figures. From this figure, the proposed iQGME-VB method is robust for all the experimental settings under both performance criteria.

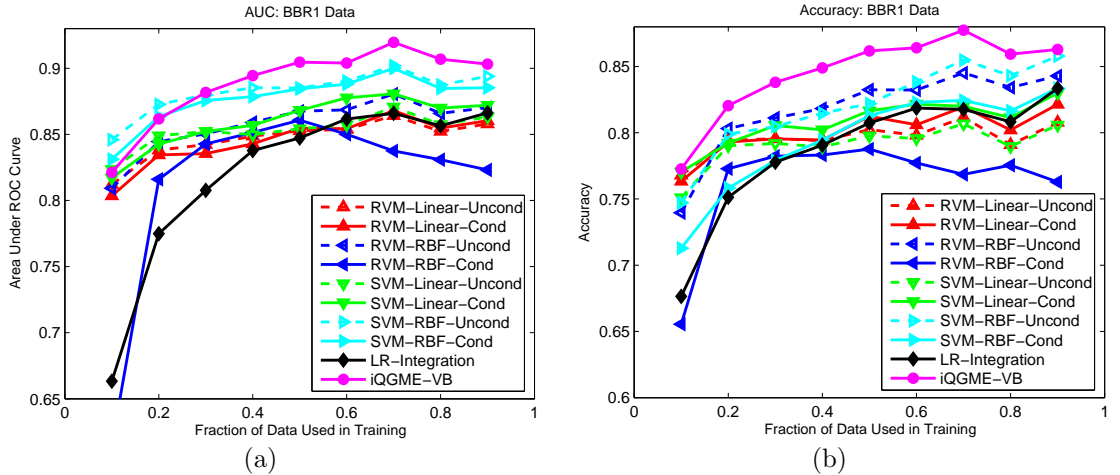


FIGURE 3.12: Mean performance over 100 random training/test partitions for each training fraction on the unexploded ordnance data set, in terms of (a) area under the ROC curve, and (b) classification accuracy.

3.5.4 Sepsis classification data

In Sections 3.5.2 and 3.5.3, we have demonstrated the proposed iQGME-VB on data sets with low to moderate dimensionality. A high-dimensional data set with natural missing values is considered in this subsection. These data were made available to the authors from the National Center for Genomic Research in the US, and will be made available upon request. This is another example for which missing data are a natural consequence of the sensing modality. There are 121 patients who are infected by sepsis, with 90 of them surviving (label -1) and 31 of them who die (label 1). For each patient, we have 521 metabolic features and 100 protein features. The purpose is to predict whether a patient infected by sepsis will die given his/her features. The missing pattern of feature values is shown in Figure 3.13(a), where black indicates observed (this missingness is a natural consequence of the sensing device).

As the data are in a 621-dimensional feature space, with only 121 samples available, we use the MFA-based variant of the iQGME (Section 3.2.3). To impose the low-rank belief for each cluster, we set $c_0 = d_0 = 1$, and the largest possible dimen-

sionality for clusters is set to be $L = 50$.

We compare to the same algorithms considered in Section 3.5.3, except the LR-Integration algorithm since it is not capable of handling such a high-dimensional data set. Mean AUC over ten random partitions are reported in Figure 3.13(b). Here we report the SVM and RVM results on the original data since they are able to classify the data in the original 621-dimensional space after missing values are imputed; we also examined SVM and RVM results on the data in a lower-dimensional latent space, after first performing factor analysis on the data, and these results were very similar to the SVM/RVM results in the original 621-dimensional space. From Figure 3.13(b), our method provides improvement by handling missing values analytically in the procedure of model inference and performing a dimensionality reduction jointly with local classifiers learning.

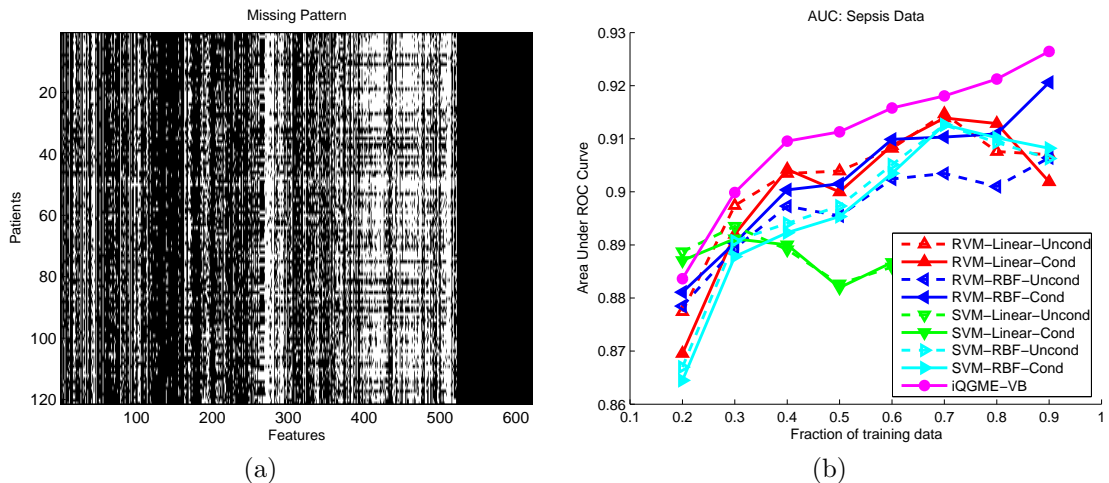


FIGURE 3.13: Sepsis data set. (a) Missing pattern, where black and white indicate observed and missing, respectively, (b) mean performance over 100 random training/test partitions for each training fraction.

3.6 Summary

This chapter proposes a classifier capable of handling incomplete data within non-parametric Bayesian framework. Features and labels are modeled jointly as mixture of experts, with the number of experts automatically inferred via the imposing of a Dirichlet process prior. With multiple simple linear experts, the proposed classifier is able to address problems with nonlinear class boundaries, and missing features could be handled in the process of model inference. In the non-parametric Bayesian framework, the uncertainty on the model structure (number of experts) and the uncertainty on the model parameters are both captured to enhance the model ability of generalization; and the hierarchical model structure makes it amenable to be extended to a multi-task setting. Fast inference is implemented via variational Bayesian.

After an illustration is given on a synthetic data set, the robustness and the performance improvement of the proposed classifier is validated on benchmark data sets, compared to previous work and two state-of-the-art classifiers. Moreover, two real applications with natural missing features are considered.

Multi-task Classification with Incomplete Data

4.1 Introduction

In addition to challenges with incomplete data, one must often address an insufficient quantity of training data. In [WLX⁺07] the authors employed semi-supervised learning [Zhu05] to address this challenge, using the contextual information in the unlabeled data to augment the limited labeled data, all done in the presence of missing/incomplete data. Another form of context one may employ to address limited labeled data is multi-task learning (MTL) [Car97, AZ05], which allows the learning of multiple tasks simultaneously to improve generalization performance. The work of [Car97] provided an overview of MTL and demonstrated it on multiple problems. In recent research, a hierarchical statistical structure has been favored for such models, where information is transferred via a common prior within a hierarchical Bayesian model [YST⁺03, ZGY06]. Specifically, information may be transferred among related tasks [XLCK07] when the Dirichlet process (DP) [Fer73] is introduced as a common prior. To the best of our knowledge, there is no previous example of addressing incomplete data in a multi-task setting, this problem constituting an important as-

pect of this dissertation. In this chapter the infinite quadratically gated mixture of experts (iQGME) algorithm is further extended to a multi-task setting, again using a non-parametric Bayesian model, simultaneously learning J missing-data classification problems, with appropriate sharing (could be global or local in feature space).

4.2 Multi-Task Learning via the Hierarchical Dirichlet Process

Assume we have J data sets, with the j th represented as $\mathcal{D}_j = \{(\mathbf{x}_{ji}, y_{ji}) : i = 1, \dots, n_j\}$; our goal is to design a classifier for each data set, with the design of each classifier termed a “task”. One may learn separate classifiers for each of the J data sets (single-task learning) by ignoring connections between the data sets, or a single classifier may be learned based on the union of all data (pooling) by ignoring differences between the data sets. More appropriately, in a hierarchical Bayesian framework J task-dependent classifiers may be learned jointly, with information borrowed via a higher-level prior (multi-task learning). In some previous research all tasks are assumed to be equally related to each other [YST⁺03, ZGY06], or related tasks share exactly the same task-dependent classifier [XLCK07]. With multiple local experts, the proposed iQGME model for a particular task is relatively flexible, enabling the borrowing of information across the J tasks (two data sets may share *parts* of the respective classifiers, without requiring sharing of all classifier components).

As discussed in Chapter 2, a DP prior encourages clustering (each cluster corresponds to a mixture component or a local expert). Now considering multiple tasks, a hierarchical Dirichlet process (HDP) [TMIJB06] may be considered to solve the problem of sharing clusters (local experts) across multiple tasks. Assume a random measure G_j is associated with each task j , where each G_j is an independent draw from Dirichlet process $\mathcal{DP}(\alpha G_0)$ with a base measure G_0 drawn from an upper-level

Dirichlet process $\mathcal{DP}(\beta H)$, *i.e.*,

$$\begin{aligned} G_j &\sim \mathcal{DP}(\alpha G_0), \text{ for } j = 1, \dots, J, \\ G_0 &\sim \mathcal{DP}(\beta H). \end{aligned}$$

As a draw from a Dirichlet process, G_0 is discrete with probability one and has a stick-breaking representation as in (2.2). With such a base measure, the task-dependent DPs reuse the atoms $\boldsymbol{\theta}_h^*$ defined in G_0 , yielding the desired sharing of atoms among tasks.

With the task-dependent iQGME defined in (3.7), we consider all J tasks jointly:

$$\begin{aligned} (\mathbf{x}_{ji}, t_{ji}) &\sim \mathcal{N}_P(\mathbf{x}_{ji} | \boldsymbol{\mu}_{ji}, \boldsymbol{\Lambda}_{ji}^{-1}) \mathcal{N}(t_{ji} | \mathbf{w}_{ji}^T \mathbf{x}_{ji}^b, 1), \\ (\boldsymbol{\mu}_{ji}, \boldsymbol{\Lambda}_{ji}, \mathbf{w}_{ji}) &\stackrel{iid}{\sim} G_j, \\ G_j &\sim \mathcal{DP}(\alpha G_0), \\ G_0 &\sim \mathcal{DP}(\beta H). \end{aligned}$$

In this form of borrowing information, experts with associated means and precision matrices are shared across tasks as distinct atoms. Since means and precision matrices statistically define local regions in feature space, sharing is encouraged locally. We explicitly write the stick-breaking representations for G_j and G_0 , with z_{ji} and c_{jh} introduced as the indicators for each data point and each distinct atom of G_j , respectively. By factorizing the base measure H as a product of a normal-Wishart prior for $(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s)$ and a normal prior for \mathbf{w}_s , the hierarchical model of the multi-task iQGME via the HDP is represented as

Data Generation:

$$\begin{aligned} (t_{ji} | c_{jh} = s, z_{ji} = h) &\sim \mathcal{N}(\mathbf{w}_s^T \mathbf{x}_{ji}^b, 1), \\ (\mathbf{x}_{ji} | c_{jh} = s, z_{ji} = h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s^{-1}), \end{aligned}$$

Drawing lower-level indicators:

$$z_{ji} \sim \sum_{h=1}^{\infty} \pi_{jh} \delta_h, \quad \text{where } \pi_{jh} = V_{jh} \prod_{l<h} (1 - V_{jl}),$$

$$V_{jh} \sim Be(1, \alpha),$$

Drawing upper-level indicators:

$$c_{jh} \sim \sum_{s=1}^{\infty} \eta_s \delta_s, \quad \text{where } \eta_s = U_s \prod_{l<s} (1 - U_l),$$

$$U_s \sim Be(1, \beta),$$

Drawing parameters from H :

$$(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s) \sim \mathcal{N}_P(\boldsymbol{\mu}_s | \mathbf{m}_0, u_0^{-1} \boldsymbol{\Lambda}_s^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_s | \mathbf{B}_0, \nu_0),$$

$$\mathbf{w}_s \sim \mathcal{N}_{P+1}(\boldsymbol{\zeta}, [\text{diag}(\boldsymbol{\lambda})]^{-1}).$$

where $j = 1, \dots, J$ and $i = 1, \dots, n_j$ index tasks and data points in each tasks, respectively; $h = 1, \dots, \infty$ and $s = 1, \dots, \infty$ index atoms for task-dependent G_j and the globally shared base G_0 , respectively. Hyper-priors are imposed similarly as in the single-task case:

$$\alpha \sim Ga(\tau_{10}, \tau_{20}),$$

$$\beta \sim Ga(\tau_{30}, \tau_{40}),$$

$$(\boldsymbol{\zeta} | \boldsymbol{\lambda}) \sim \mathcal{N}_{P+1}(\mathbf{0}, \gamma_0^{-1} [\text{diag}(\boldsymbol{\lambda})]^{-1}),$$

$$\lambda_p \sim Ga(a_0, b_0), \quad p = 1, \dots, P + 1,$$

The graphical representation of the iQGME for multi-task learning via the HDP is shown in Figure 4.1.

4.3 Variational Bayesian Inference

For multi-task learning much of the inference is highly related to that of single-task learning, as discussed in Chapter 3; in the following we focus only on differences.

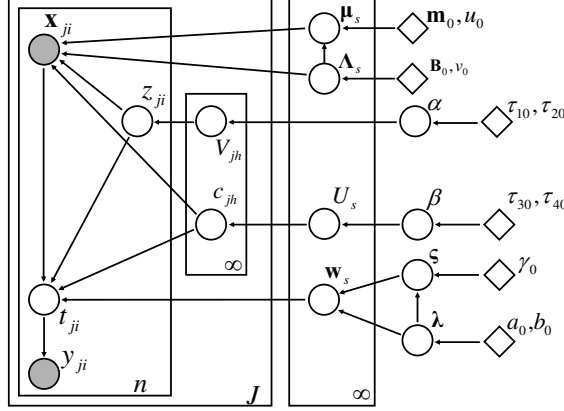


FIGURE 4.1: Graphical representation of the iQGME for multi-task learning via the hierarchical Dirichlet process (HDP). Refer to Figure 3.1 for additional information.

In the multi-task learning model, we denote the collection of latent variables as $\Theta = \{\mathbf{t}, \mathbf{x}^m, \mathbf{z}, \mathbf{V}, \alpha, \mathbf{c}, \mathbf{U}, \beta, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}\}$, and the collection of hyper-parameters as $\Psi = \{\mathbf{m}_0, u_0, \mathbf{B}_0, \nu_0, \tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}, \gamma_0, a_0, b_0\}$. The factorized variational distributions are specified as

$$\begin{aligned}
& q(\mathbf{t}, \mathbf{x}^m, \mathbf{z}, \mathbf{V}, \alpha, \mathbf{c}, \mathbf{U}, \beta, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}) \\
&= \prod_{j=1}^J \left\{ \prod_{i=1}^{n_j} [q(t_{ji})q(\mathbf{x}_{ji}^m)q(z_{ji})] \prod_{h=1}^{N-1} q(V_{jh}) \prod_{h=1}^N q(c_{jh}) \right\} \prod_{s=1}^{S-1} q(U_s) \\
& \quad \prod_{s=1}^S [q(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s)q(\mathbf{w}_s)] \prod_{p=1}^{P+1} q(\zeta_p, \lambda_p)q(\alpha)q(\beta).
\end{aligned}$$

The variational distributions of $(t_{ji}, V_{jh}, \alpha, \boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s, \mathbf{w}_s, \zeta_p, \lambda_p)$ are assumed to be the same as in the single-task learning, while the variational distributions of hidden variables newly introduced for the upper-level Dirichlet process are specified as

- $q(c_{jh})$ for each indicator c_{jh} is a multinomial distribution with probabilities $\boldsymbol{\sigma}_{jh}$,

$$c_{jh} \sim \mathcal{M}_S(1, \sigma_{jh1}, \dots, \sigma_{jhS}), \quad j = 1, \dots, J, \quad h = 1, \dots, N.$$

- $q(U_s)$ for each weight U_s is a Beta distribution,

$$U_s \sim Be(\kappa_{s1}, \kappa_{s2}), \quad s = 1, \dots, S-1.$$

Here we have a truncation level of S for the upper-level DP, which implies that the mixture proportions $\eta_s(\mathbf{U})$ are equal to zero for $s > S$. Therefore, $q(U_s) = \delta_1$ for $s = S$, and $q(U_s) = \delta_0$ for $s > S$. For $s < S$, U_s has a variational Beta posterior.

- $q(\beta)$ for the scaling parameter β is a Gamma distribution,

$$\beta \sim Ga(\tau_3, \tau_4).$$

We also note that with a higher-level of hierarchy, the dependency between the missing values $\mathbf{x}_{ji}^{m_{ji}}$ and the associated indicator z_{ji} has to be broken so that the inference becomes tractable. The variational distribution of z_{ji} is still assumed to be multinomial distributed, while $\mathbf{x}_{ji}^{m_{ji}}$ is assumed to be normally distributed but no longer dependent on z_{ji} .

The update equations of multi-task learning with incomplete data are summarized as follows:

1. $q(t_{ji} | \mu_{ji}^t)$

$$\mu_{ji}^t = \sum_{s=1}^S E\sigma_{jis} \langle \mathbf{w}_s \rangle^T \hat{\mathbf{x}}_{ji}^b \quad \text{where} \quad E\sigma_{jis} = \sum_{h=1}^N \rho_{jih} \sigma_{jhs}, \quad \hat{\mathbf{x}}_{ji}^b = [\mathbf{x}_{ji}^{o_{ji}}; \mathbf{m}_{ji}^{m_{ji}|o_{ji}}; 1].$$

2. $q(\mathbf{x}_{ji}^{m_{ji}} | \mathbf{m}_{ji}^{m_{ji}|o_{ji}}, \Sigma_{ji}^{m_{ji}|o_{ji}})$

$$\begin{aligned} \mathbf{m}_{ji}^{m_{ji}|o_{ji}} &= \tilde{\boldsymbol{\mu}}_{ji}^{m_{ji}} + \tilde{\Sigma}_{ji}^{m_{ji}o_{ji}} (\tilde{\Sigma}_{ji}^{o_{ji}o_{ji}})^{-1} (\mathbf{x}_{ji}^{o_{ji}} - \tilde{\boldsymbol{\mu}}_{ji}^{o_{ji}}), \\ \Sigma_{ji}^{m_{ji}|o_{ji}} &= \tilde{\Sigma}_{ji}^{m_{ji}m_{ji}} - \tilde{\Sigma}_{ji}^{m_{ji}o_{ji}} (\tilde{\Sigma}_{ji}^{o_{ji}o_{ji}})^{-1} \tilde{\Sigma}_{ji}^{o_{ji}m_{ji}}, \end{aligned}$$

where

$$\tilde{\Sigma}_{ji} = \left(\sum_{s=1}^S E\sigma_{jis} (\langle \mathbf{w}_s^x (\mathbf{w}_s^x)^T \rangle + \nu_s \mathbf{B}_s) \right)^{-1},$$

$$\tilde{\boldsymbol{\mu}}_{ji} = \tilde{\Sigma}_{ji} \sum_{s=1}^S E\sigma_{jis} (\langle t_{ji} \rangle \langle \mathbf{w}_s^x \rangle + \nu_s \mathbf{B}_s \mathbf{m}_s - \langle \mathbf{w}_s^x \mathbf{w}_s^b \rangle).$$

$$\hat{\mathbf{x}}_{ji} = \begin{bmatrix} \mathbf{x}_{ji}^{o_{ji}} \\ \mathbf{m}_{ji}^{m_{ji}|o_{ji}} \end{bmatrix}, \quad \hat{\boldsymbol{\Omega}}_{ji} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{ji}^{m_{ji}|o_{ji}} \end{bmatrix}, \quad \langle \mathbf{x}_{ji} \mathbf{x}_{ji}^T \rangle = \hat{\mathbf{x}}_{ji} \hat{\mathbf{x}}_{ji}^T + \hat{\boldsymbol{\Omega}}_{ji}.$$

3. $q(z_{ji} | \boldsymbol{\rho}_{ji})$

$$\begin{aligned} \rho_{jih} &= q(z_{ji} = h) \\ &\propto \exp \left\{ \sum_{s=1}^S \sigma_{jhs} [\langle t_{ji} \rangle \langle \mathbf{w}_s \rangle^T \hat{\mathbf{x}}_{ji}^b - \frac{1}{2} \text{tr}(\langle \mathbf{w}_s \mathbf{w}_s^T \rangle \langle \mathbf{x}_{ji}^b (\mathbf{x}_{ji}^b)^T \rangle)] \right. \\ &\quad \left. + \langle \ln V_{jh} \rangle + \sum_{l < h} \langle \ln(1 - V_{jl}) \rangle \right. \\ &\quad \left. + \frac{1}{2} \sum_{s=1}^S \sigma_{jhs} [\langle \ln |\boldsymbol{\Lambda}_s| \rangle - (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s)^T \nu_s \mathbf{B}_s (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s) - P/u_s - \text{tr}(\nu_s \mathbf{B}_s \hat{\boldsymbol{\Omega}}_{ji})] \right\}. \end{aligned}$$

4. $q(\mathbf{V} | \mathbf{v})$

$$v_{jh1} = 1 + \sum_{i=1}^{n_j} \rho_{jih}, \quad v_{jh2} = \langle \alpha \rangle + \sum_{l>h} \sum_{i=1}^{n_j} \rho_{jil}.$$

$$\langle \ln V_{jh} \rangle = \psi(v_{jh1}) - \psi(v_{jh1} + v_{jh2}), \quad \langle \ln(1 - V_{jl}) \rangle = \psi(v_{jl2}) - \psi(v_{jl1} + v_{jl2}).$$

5. $q(\alpha | \tau_1, \tau_2), \langle \alpha \rangle = \tau_1 / \tau_2.$

$$\tau_1 = J(N-1) + \tau_{10}, \quad \tau_2 = \tau_{20} - \sum_{j=1}^J \sum_{h=1}^{N-1} \langle \ln(1 - V_{jh}) \rangle.$$

6. $q(\mathbf{c}|\boldsymbol{\sigma})$

$$\begin{aligned} \sigma_{jhs} &\propto \exp\left\{\sum_{i=1}^{n_j} \rho_{jih} [\langle t_{ji} \rangle \langle \mathbf{w}_s \rangle^T \hat{\mathbf{x}}_{ji}^b - \frac{1}{2} \text{tr}(\langle \mathbf{w}_s \mathbf{w}_s^T \rangle \langle \mathbf{x}_{ji}^b (\mathbf{x}_{ji}^b)^T \rangle)] \right. \\ &\quad \left. + \langle \ln U_s \rangle + \sum_{l < s} \langle \ln(1 - U_s) \rangle \right. \\ &\quad \left. + \frac{1}{2} \sum_{i=1}^{n_j} \rho_{jih} [\langle \ln |\boldsymbol{\Lambda}_s| \rangle - (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s)^T \nu_s \mathbf{B}_s (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s) - P/u_s - \text{tr}(\nu_s \mathbf{B}_s \hat{\boldsymbol{\Omega}}_{ji})] \right\}. \end{aligned}$$

7. $q(U_s | \kappa_{s1}, \kappa_{s2})$

$$\kappa_{s1} = 1 + \sum_{j=1}^J \sum_{h=1}^N \sigma_{jhs}, \quad \kappa_{s2} = \langle \beta \rangle + \sum_{j=1}^J \sum_{h=1}^N \sum_{l > s} \sigma_{jhl}.$$

$$\langle \ln U_s \rangle = \psi(\kappa_{s1}) - \psi(\kappa_{s1} + \kappa_{s2}), \quad \langle \ln(1 - U_s) \rangle = \psi(\kappa_{s2}) - \psi(\kappa_{s1} + \kappa_{s2}).$$

8. $q(\beta | \tau_3, \tau_4), \langle \beta \rangle = \tau_3 / \tau_4$.

$$\tau_3 = S - 1 + \tau_{30}, \quad \tau_4 = \tau_{40} - \sum_{s=1}^{S-1} \langle \ln(1 - U_s) \rangle.$$

9. $q(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s | \mathbf{m}_s, u_s, \mathbf{B}_s, \nu_s)$

$$\nu_s = \nu_0 + N_s, \quad u_s = u_0 + N_s, \quad \mathbf{m}_s = \frac{u_0 \mathbf{m}_0 + N_s \bar{\mathbf{x}}_s}{u_s},$$

$$\mathbf{B}_s^{-1} = \mathbf{B}_0^{-1} + \sum_{j=1}^J \sum_{i=1}^{n_j} E\sigma_{jis} \hat{\boldsymbol{\Omega}}_{ji} + N_s \bar{\mathbf{S}}_s + \frac{u_0 N_s}{u_s} (\bar{\mathbf{x}}_s - \mathbf{m}_0)(\bar{\mathbf{x}}_s - \mathbf{m}_0)^T,$$

where $E\sigma_{jis} = \sum_{h=1}^N \rho_{jih} \sigma_{jhs}$, and

$$N_s = \sum_{j=1}^J \sum_{i=1}^{n_j} E\sigma_{jis}, \quad \bar{\mathbf{x}}_s = \sum_{j=1}^J \sum_{i=1}^{n_j} E\sigma_{jis} \hat{\mathbf{x}}_{ji} / N_s,$$

$$\bar{\mathbf{S}}_s = \sum_{j=1}^J \sum_{i=1}^{n_j} E\sigma_{jis} (\hat{\mathbf{x}}_{ji} - \bar{\mathbf{x}}_s)(\hat{\mathbf{x}}_{ji} - \bar{\mathbf{x}}_s)^T / N_s.$$

$$\langle \ln |\mathbf{\Lambda}_s| \rangle = \sum_{p=1}^P \psi((\nu_s - p + 1)/2) + P \ln 2 + \ln |\mathbf{B}_s|,$$

$$\langle (\mathbf{x}_{ji} - \boldsymbol{\mu}_s)^T \mathbf{\Lambda}_s (\mathbf{x}_{ji} - \boldsymbol{\mu}_s) \rangle = (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s)^T \nu_s \mathbf{B}_s (\hat{\mathbf{x}}_{ji} - \mathbf{m}_s) + P/u_s + \text{tr}(\nu_s \mathbf{B}_s \hat{\mathbf{\Omega}}_{ji}).$$

10. $q(\mathbf{w}_s | \boldsymbol{\mu}_s^w, \boldsymbol{\Sigma}_s^w)$

$$\boldsymbol{\Sigma}_s^w = \left(\sum_{j=1}^J \sum_{i=1}^{n_j} E \sigma_{jis} (\hat{\mathbf{x}}_{ji}^b \hat{\mathbf{x}}_{ji}^{bT} + \hat{\mathbf{\Omega}}_{ji}) + \text{diag}(\langle \boldsymbol{\lambda} \rangle) \right)^{-1},$$

$$\boldsymbol{\mu}_s^w = \boldsymbol{\Sigma}_s^w \left(\sum_{j=1}^J \sum_{i=1}^{n_j} E \sigma_{jis} \hat{\mathbf{x}}_{ji}^b \langle t_{ji} \rangle + \text{diag}(\langle \boldsymbol{\lambda} \rangle) \phi \right).$$

$$\langle \mathbf{w}_s \rangle = \boldsymbol{\mu}_s^w, \quad \langle \mathbf{w}_s \mathbf{w}_s^T \rangle = \boldsymbol{\Sigma}_s^w + \boldsymbol{\mu}_s^w (\boldsymbol{\mu}_s^w)^T.$$

11. $q(\lambda_p | a_p, b_p), \langle \lambda_p \rangle = a_p / b_p$.

$$\phi_p = \sum_{s=1}^S \langle w_{sp} \rangle / \gamma, \quad \gamma = \gamma_0 + S,$$

$$a_p = a_0 + \frac{S}{2}, \quad b_p = b_0 + \frac{1}{2} \sum_{s=1}^S \langle W_{sp}^2 \rangle - \frac{1}{2} \gamma \phi_p^2.$$

4.4 Experimental Results

As stated in Chapter 3, the multi-task model is also insensitive to the setting of hyper-parameters of priors. We maintain the hyper-parameter settings as used for the single-task learning (again, no tuning has been performed). New hyper-parameters defining the higher-level DP precision γ are set as those for the lower-level DPs, i.e., $\tau_{30} = 0.05$ and $\tau_{40} = 0.05$. The truncation levels for the variational distributions are set to be $N = 20$ and $S = 50$.

The initialization is also similar to that for the single-task setting. Most variational hyper-parameters are initialized using the corresponding prior hyper-parameters,

which are data-independent. The precision/covariance matrices \mathbf{B}_s and Σ_s^w are simply initialized as identity matrices. However, the variational mean of the soft label μ_{ji}^t is initialized by the associated label y_{ji} . The initial values for the variational mean of the Gaussian means \mathbf{m}_s , the sample indicator probabilities ρ_{ji} , and the cluster indicator probabilities σ_{jh} are set as if in a single-task setting, so that the candidates for \mathbf{m}_s are diverse enough to describe the data samples from each task in feature space. When the algorithm figures out that some of the components could be presented with the same parameters, data associated with those components (may from different tasks) will merge together. Specifically, a K-means clustering algorithm is implemented on the feature vectors within each task. The cluster means for all the tasks are concatenated to initialize the variational mean of the Gaussian means \mathbf{m}_s . The cluster identifications for data samples are used to initialize the indicator probabilities ρ_{ji} . Accordingly, the cluster indicator probabilities σ_{jh} are initialized with the task identifications for clusters. It is believed that this initialization gives a reasonably good start for the algorithm since it represents a tendency to learn components in each task individually. Performance for individual task will not be dramatically influenced if they are not quite similar; while components could become shared between tasks when they are really related.

4.4.1 Landmine Detection Data

In an application of landmine detection, data collected from 19 landmine fields are treated as 19 subtasks (available at <http://www.ee.duke.edu/lcarin/LandmineData.zip>). Among them, subtasks 1-10 correspond to regions that are relatively highly foliated and subtasks 11-19 correspond to regions that are bare earth or desert. In all subtasks, each target is characterized by a 9-dimensional feature vector \mathbf{x} with corresponding binary label y (1 for landmines and -1 for clutter). The number of landmines and clutter in each task is summarized in Figure 4.2. The feature vec-

tors are extracted from images measured with airborne radar systems. A detailed description of this landmine data set has been presented elsewhere [XLCK07].

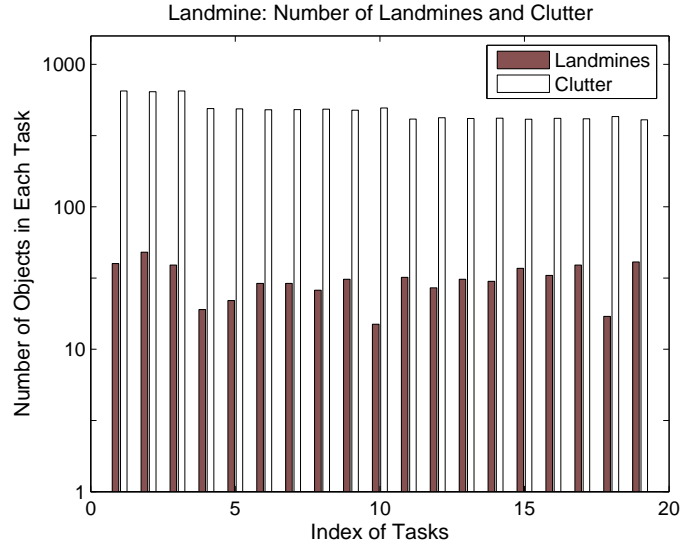


FIGURE 4.2: Number of landmines and clutter in each task for the landmine-detection data set [XLCK07].

Although our main objective is to simultaneously learn classifiers for multiple tasks with incomplete data, we first demonstrate the proposed iQGME-based multi-task learning (MTL) model on the complete data, comparing it to two multi-task learning algorithms designed for the situation with all the features observed. One is based on task-specific logistic regression (LR) models, with the DP as a hierarchical prior across all the tasks [XLCK07]; the other assumes an underlying structure, which is shared by all the tasks [AZ05]. For the LR-MTL algorithm, we cite results on complete data from [XLCK07], and implement the authors’ Matlab code with default hyper-parameters on the cases with incomplete data. The Matlab implementation for the Structure-MTL algorithm is included in the “Transfer Learning Toolkit for Matlab” available at <http://multitask.cs.berkeley.edu/>. The dimension of the underlying structure is a user-set parameter, and it should be smaller than the

feature dimension in the original space. As the dimension of the landmine detection data is 9, we set the hidden dimension as 5. We also tried 6, 7, and 8, and did not observe big differences in performance. Single-task learning (STL) iQGME and LR models are also included for comparison.

Each task is divided into training and test subsets randomly. Since the number of elements in the two classes is highly unbalanced, as shown in Figure 4.2, we impose that there is at least one instance from each class in each subset. Following [XLCK07], the size of the training subset in each task varies from 20 to 300 in increments of 20, and 100 independent trials are performed for each size of data set. An average AUC [HM82] over all the 19 tasks is calculated as the performance representation for one trial of a given training size. Results are reported in Figure 4.3.

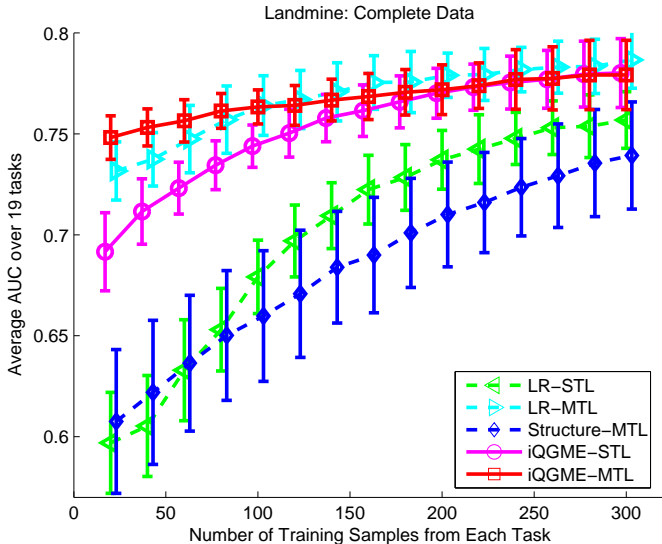


FIGURE 4.3: Average AUC over 19 tasks of landmine detection with complete data. Error bars reflect the standard deviation across 100 random partitions of training and test subsets. Results of logistic regression based algorithms are cited from [XLCK07], where LR-MTL and LR-STL respectively correspond to SMTL-2 and STL in Figure 3 of [XLCK07].

The first observation from Figure 4.3 is that we obtain a significant performance improvement for single-task learning by using the iQGME-VB instead of the linear

logistic regression model [XLCK07]. We also notice that the multi-task algorithm based on iQGME-VB further improves the performance when the training data are scarce, and yields comparable overall results as the LR-MTL does. The structure-MTL does not perform well on this data set. We suspect that a hidden structure in such a 9-dimensional space does not necessarily exist. Another possible reason may be that the minimization of empirical risk is sensitive for the cases with highly unbalanced labels, as for this data set.

It is also interesting to explore the similarity between tasks. The similarity defined by different algorithms may be different. In [XLCK07], two tasks are defined to be similar if they share the same linear classifier. However, with the joint distribution of covariates and the response, the iQGME-MTL requires both the data distributions and the classification boundaries to be similar if two tasks are deemed to be similar. Another difference is that two tasks could be partially similar since sharing between tasks is encouraged at the cluster-level instead of at the task-level ([XLCK07] employs task-level clustering). We generate the similarity matrices between tasks as follows: In each random trial, there are in total S higher-level items shared among tasks. For each task, we can find the task-specific probability mass function (pmf) over all the higher-level items. Using these pmfs as the characteristics for tasks in the current trial, we calculate the pair-wise Kullback-Leibler (KL) distances and convert them to similarity measures through a minus exponential function. Results of multiple trials are summed over and normalized as shown in Figure 4.4. It can be seen that the similarity structure among tasks becomes clearer when we have more training data available. As discovered by [XLCK07], we also find two big clusters correspond to two different vegetation conditions of the landmine fields (task 1-10 and task 11-19). Further sub-structures among tasks are also explored by the iQGME-MTL model, which may suggest other unknown difference among the landmine fields.

After yielding competitive results on the landmine-detection data set with com-

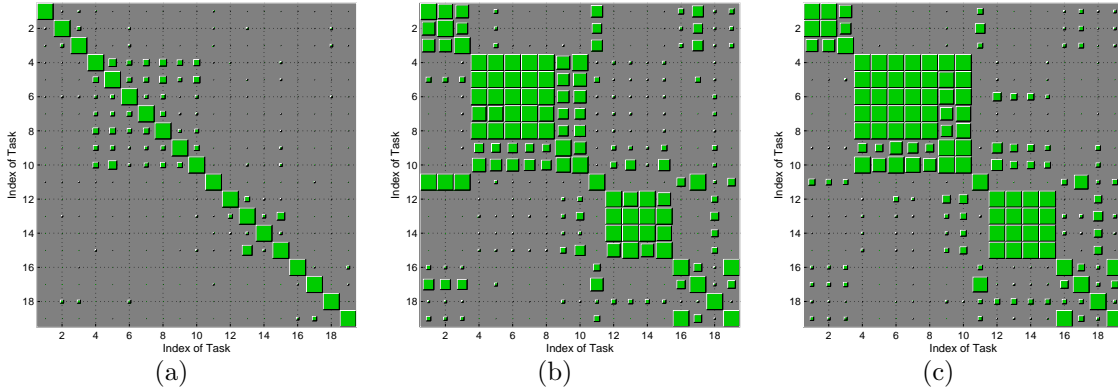


FIGURE 4.4: Similarity between tasks in the landmine detection problem with complete data given (a) 20, (b) 100, and (c) 300 training samples from each task. The size of green blocks represent the value of the corresponding matrix element.

plete data, the iQGME-based algorithms are evaluated on incomplete data, which are simulated by randomly removing a portion of feature values for each task as in Section 3.5.2. We consider three different portions of missing values: 25%, 50% and 75%. As in the experiments above on benchmark data sets, we perform ten independent random trials for each setting of missing fraction and training size.

To the best of our knowledge, there exists no previous work in the literature on multi-task learning with missing data. As presented in Figure 4.5, we use the LR-MTL [XLCK07] and the Structure-MTL [AZ05] with missing values imputed as baseline algorithms. Results of the two-step LR with integration [WLX⁺07] and the LR-STL with single imputations are also included for comparison. Imputations using both unconditional-means and conditional-means are considered. From Figure 4.5, iQGME-STL consistently performs best among single-task learning methods and even better than LR-MTL-Uncond when the size of the training set is relatively large. The imputations using conditional-means yields consistently better results for the LR-based models on this data set. The iQGME-MTL outperforms the baselines and all the single-task learning methods overall. Furthermore, the improvement of iQGME-MTL is more pronounced when there are more features missing. These ob-

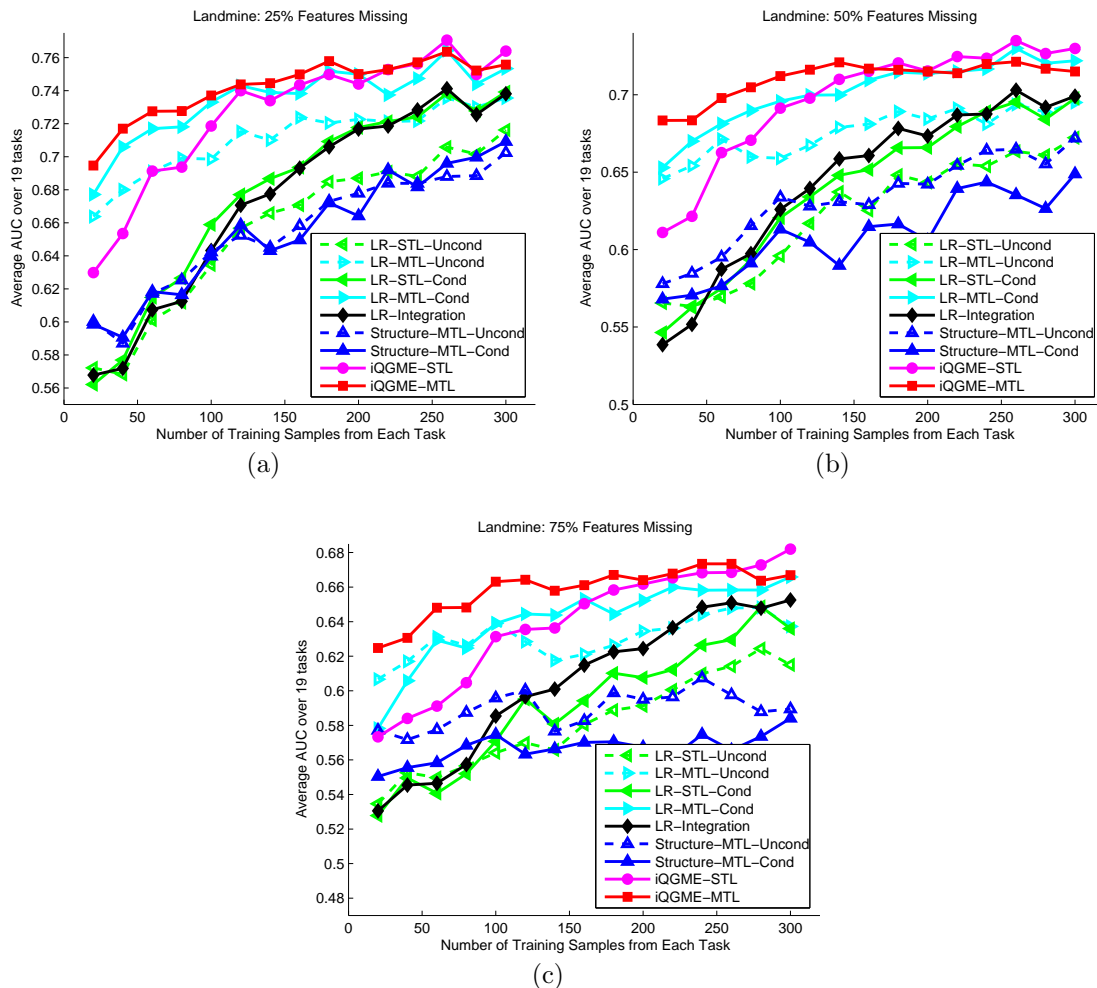


FIGURE 4.5: Average AUC over 19 tasks of landmine detection for the cases when (a) 25%, (b) 50%, and (c) 75% of the features are missing. Mean values of performance across 10 random partitions of training and test subsets are reported. Error bars are omitted for legibility.

servations underscore the advantage of handling missing data in a principled manner and at the same time learning multiple tasks simultaneously.

The task-similarity matrices for the incomplete-data cases are shown in Figure 4.6. It can be seen that when a small fraction (*e.g.*, 25%) of the feature values are missing and training data are rich (*e.g.*, 300 samples from each task), the similarity pattern among tasks is similar to what we have seen for the complete-data case. As the fraction of missing values becomes larger, tasks appear more different from each

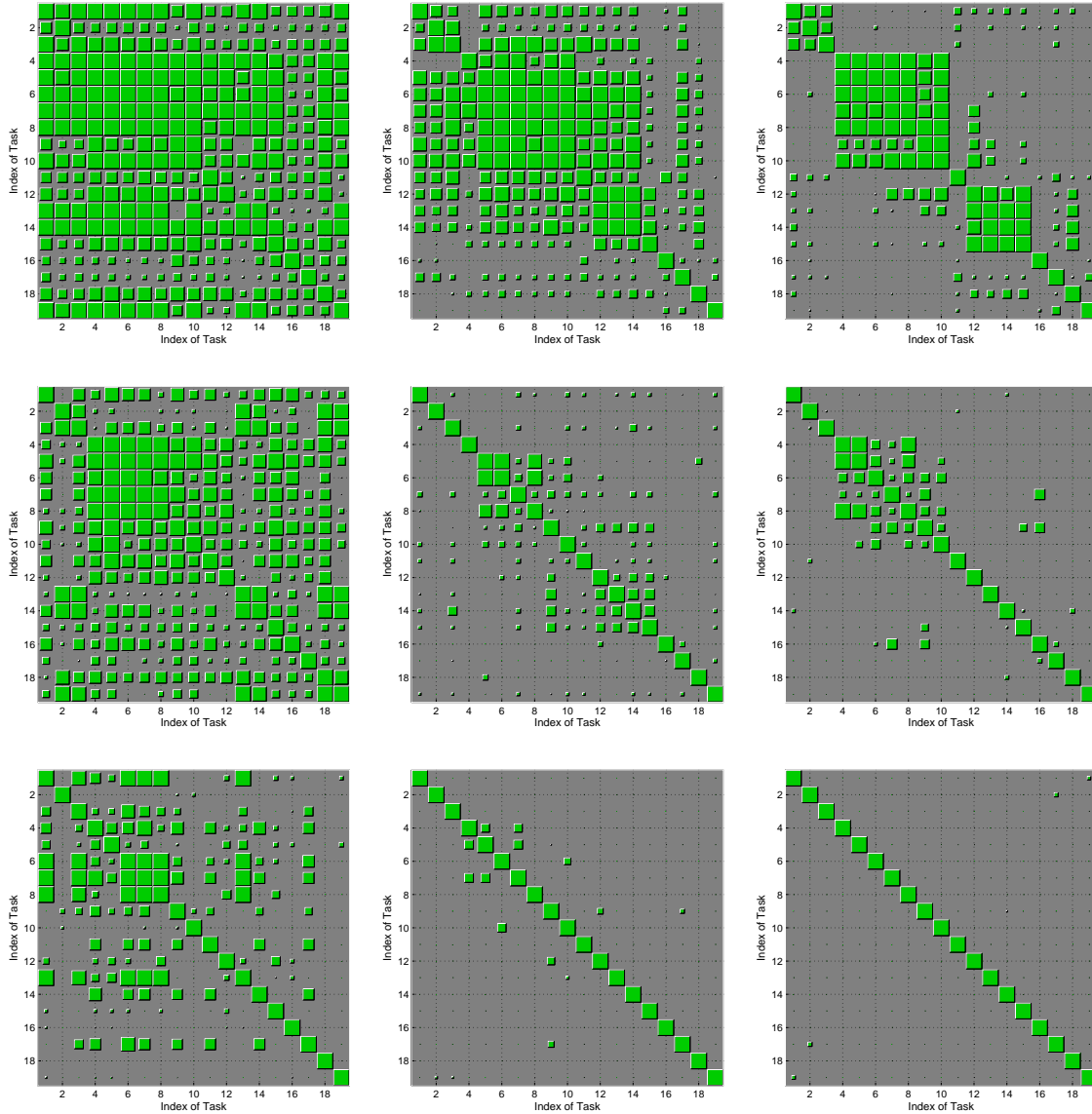


FIGURE 4.6: Similarity between tasks in the landmine detection problem with incomplete data. Row 1, 2 and 3 corresponds to the cases with 25%, 50% and 75% features missing, respectively; column 1, 2 and 3 corresponds to the cases with 20, 100 and 300 training samples from each task, respectively.

other in terms of the usage of the higher-level items. Considering that the missing pattern for each task is unique, it is probable that tasks look quite different from each other after a large fraction of feature values are missing. However, the fact that tasks tend to use different subsets of higher-level items does not mean it is equivalent

to learning them separately (STL), as parameters of the common base measures are inferred based on all the tasks.

4.4.2 Handwritten letters data

Another example corresponds to multi-task learning of classifiers for handwritten letters, this data set included in the “Transfer Learning Toolkit for Matlab” available at <http://multitask.cs.berkeley.edu/>. The objective of each task is to distinguish two letters which are easily confused. The number of samples for all the letters considered in the total eight tasks is summarized in Table 4.1. Each sample is a 16×8 image as shown in Figure 4.7. We use the 128 pixel values of each sample directly as its feature vector.

Table 4.1: Handwritten letters classification data set.

| Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 | Task 7 | Task 8 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| ‘c’: 2107 | ‘g’: 2460 | ‘m’: 1596 | ‘a’: 4016 | ‘i’: 4895 | ‘a’: 4016 | ‘f’: 918 | ‘h’: 858 |
| ‘e’: 4928 | ‘y’: 1218 | ‘n’: 5004 | ‘g’: 2460 | ‘j’: 188 | ‘o’: 3880 | ‘t’: 2131 | ‘n’: 5004 |

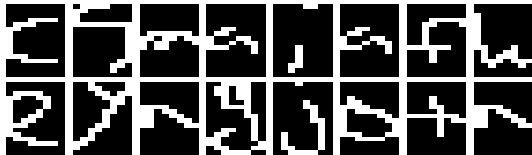


FIGURE 4.7: Sample images of the handwritten letters. The two images in each column represents the two classes in the corresponding task described in Table 4.1.

We compare the proposed iQGME-MTL algorithm to the LR-MTL [XLCK07] and the Structure-MTL [AZ05] mentioned in Section 4.4.1. For the non-parametric Bayesian methods (iQGME-MTL and LR-MTL), we use the same parameter setting as before. The dimension of the underlying structure for the Structure-MTL is set to be 50 in the results shown in Figure 4.8. We also tried 10, 20, 40, 60, 80 and 100, and did not observe big difference. From Figure 4.8, the iQGME-MTL performs significantly better than the baselines on this data set for all the missing fractions and

training fractions under consideration. As we expected, the Structure-MTL yields comparable results as the LR-MTL on this data set.

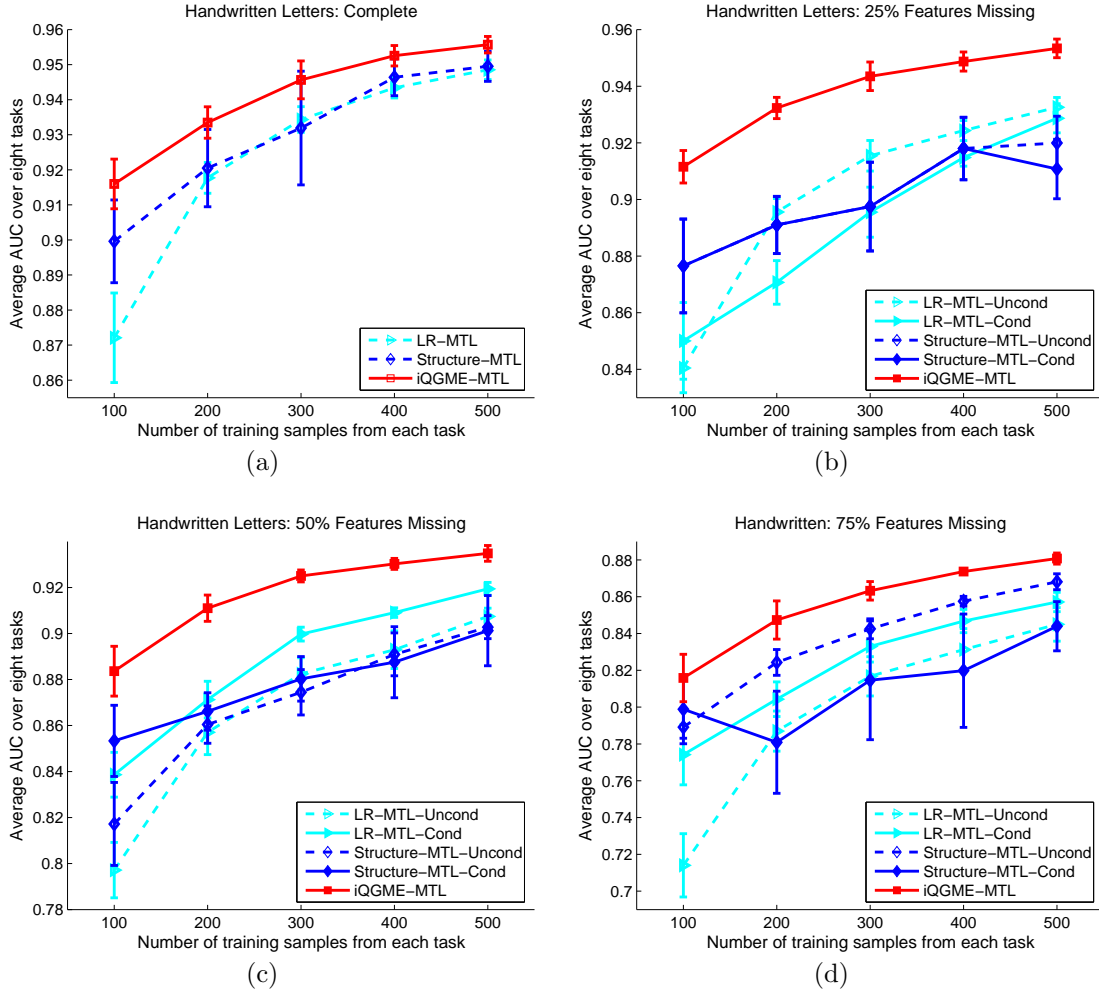


FIGURE 4.8: Average AUC over eight tasks of handwriting letters classification for the cases when (a) none, (b) 25%, (c) 50%, and (d) 75% of the features are missing. Mean values of performance with one standard deviation across 10 random partitions of training and test subsets are reported.

We also check the similarity matrices generated as described in Section 4.4.1. All eight tasks look distinct for all the experimental settings. Similar phenomena are observed for the LR-MTL. These observations are quite reasonable considering the dimensionality of the problem (128). It is almost impossible for two tasks to share exactly the same classification boundary and the same inputs distribution in a

128-dimensional space. However, as we stated in Section 4.4.1, we can benefit from learning multiple tasks simultaneously even in such a situation, since parameters of the common base measures are inferred based on all the tasks.

4.5 Summary

We extend the classifier developed in Chapter 3 to the multi-task setting in a hierarchical Bayesian framework, so that multiple classifiers could be learned jointly. It is particularly important when training data from each task are not sufficient and incomplete as well. Compared to existing multi-task learning classifiers [XLCK07, LLC07c], the proposed algorithm allows for sharing of mixture components (Gaussian component-expert pairs) across tasks by imposing a hierarchical Dirichlet process (HDP) prior.

The effectiveness of the proposed model has been successfully demonstrated on a landmine detection problem and a handwritten letters classification problem. At the same time of yielding robust classification results, the similarity between tasks is also explored in terms of the usage of the mixture components available to all the tasks.

Bayesian Matrix Completion

5.1 Introduction

In this chapter we consider the problem of recovering a matrix from its partially observable entries, referred to as matrix completion. In general, it is impossible to recover an arbitrary matrix without any further information. An extreme example is a random matrix with entries sampled independently. Fortunately, matrices we would like to recover in many practical scenarios are usually low-rank or approximately low-rank. For example, the data matrix of user-ratings may be approximately low-rank since it is reasonable to believe that only a few factors contribute to a user's preferences; the data matrix of hitter-pitcher record could also be considered approximately low-rank since the hitting results may depend on only a small number of factors.

Candès and Recht [CR09] provided conditions for exact recovery of low-rank matrices and solved the problem by minimizing the nuclear norm (summation of singular values) with the observed entries as constraints. Alternatively, Bayesian models are desirable to handle approximately low-rank matrices by introducing statistical errors;

and the uncertainty captured by Bayesian models could provide useful information for acquiring entries actively.

We propose a Bayesian singular value decomposition (BSVD) model with the pseudo rank explicitly encouraged to be small by informative priors. As [CR09], the BSVD model is a general approach for the matrix-completion problem. In this dissertation we mainly consider its application on collaborative filtering, and accordingly generalize the model to make use of auxiliary information such as user age, gender and occupation, or movie genre, year and director, and even the locations of missing entries themselves. In some scenarios with data matrices consisting of integer counts, such as the Major League Baseball (MLB) data we examine, a probit link function is introduced to connect the observation matrices with the latent continuous matrices. As an attempt, we demonstrated on synthetic examples that acquiring entries actively according to the uncertainty on entries prediction could achieve the same prediction accuracy with a smaller number of observed entries, compared to acquiring entries randomly.

5.2 Bayesian Singular Value Decomposition

Consider a partially observed data matrix $\mathbf{Y} \in \mathfrak{R}^{I \times J}$, where entries Y_{ij} with $(i, j) \in \Omega$ are observable, and other entries are missing. The cardinality of set Ω is m . The objective is to fit \mathbf{Y} with a matrix $\mathbf{X} \in \mathfrak{R}^{I \times J}$ so that the prediction on those missing entries in \mathbf{Y} could be made. Different assumptions on the fitting matrix \mathbf{X} may lead to different solutions. Among many alternatives, low-rank constraint is often a natural choice without further information since real data matrices are approximately low-rank in many applications. To avoid the NP-hard problem of directly minimizing the rank of \mathbf{X} , subject to the constraint that the entries of \mathbf{X} and \mathbf{Y} at observed locations should be the same, Candès and Recht [CR09] chose to minimize the nuclear norm (summation of singular values) of \mathbf{X} instead. In

this section we propose a Bayesian model in a similar formulation of singular value decomposition (SVD), with the psuedo rank of \mathbf{X} imposed to be small explicitly.

Mathematically, the singular value decomposition (SVD) of a matrix \mathbf{X} of rank r is

$$\mathbf{X} = \sum_{k=1}^r s_k \mathbf{u}_k \mathbf{v}_k^T \quad (5.1)$$

where left singular vectors $\mathbf{u}_k \in \mathfrak{R}^I$, right singular vectors $\mathbf{v}_k \in \mathfrak{R}^J$, and singular values $s_k \in \mathfrak{R}^+$. For the sake of identifiability, singular values are ordered descendingly, and singular vectors are required to be orthonormal. In this way, a matrix is decomposed into components $\mathbf{u}_k \mathbf{v}_k^T$ with s_k indicating associated energy. It is well known that among all the matrices of rank D , the first D components of the SVD provides the least-square-error approximation to the matrix \mathbf{X} . In matrix completion problems, since the matrix entries are incomplete, we cannot find a low-rank approximation by employing the SVD directly. However, probabilistic pseudo-SVD models [DSG09, LT07, SM07, SM08, LU09, HBC10] could be learned in the light of observed entries.

We propose a non-parametric Bayesian approach so that the number of components is not fixed but inferred from data. In order to favor constructions with only a few components, we provide a dictionary of components consisting of a large number of candidates, and introduce binary selecting variables z_k with a zero-favor prior, i.e.,

$$\mathbf{X} = \sum_{k=1}^K (s_k z_k) \mathbf{u}_k \mathbf{v}_k^T \quad (5.2)$$

where $z_k \in \{0, 1\}$ and $z_k \sim \text{Bern}(\pi_k)$, with selecting weights $\pi_k \sim \text{Beta}(a/K, b(K - 1)/K)$. For the observed data matrix \mathbf{Y} , we expect it to be approximately low-rank with some inevitable errors. Therefore,

$$\mathbf{Y} = \mathbf{X} + \mathbf{E}$$

with the noise term $\mathbf{E} \in \Re^{I \times J}$. Each component of \mathbf{E} is drawn iid from $\mathcal{N}(0, \alpha^{-1})$, with a separate gamma hyper-prior employed for α .

Through the choice of a and b we impose our prior belief about the number of components selected (or the pseudo rank of \mathbf{X}). Specifically, by marginalizing out the vector $\{\pi_1, \dots, \pi_K\}$, one can show that the number of $\{z_k\}_{k=1, K}$ equal to one is distributed Binomial($K, a/(a + b(K - 1))$), and the expected number of ones is $aK/[a + b(K - 1)]$. As $K \rightarrow \infty$, the number of non-zero z_k is drawn from Poisson(a/b). Hence, by setting, a , b , and K , one is making explicit prior statements about the pseudo rank of \mathbf{X} , and posterior inference yields the estimated rank based on the observed data.

Redundance exists in the introduce of s_k since this scalar could be absorbed into \mathbf{u}_k and \mathbf{v}_k . We retain s_k and impose $\mathbf{u}_k \sim \mathcal{N}(0, \frac{1}{I}\mathbf{I}_I)$ and $\mathbf{v}_k \sim \mathcal{N}(0, \frac{1}{J}\mathbf{I}_J)$, where \mathbf{I}_J is a $J \times J$ identity matrix. Note that the columns of \mathbf{U} and \mathbf{V} have unit expected ℓ_2 norm, with the amplitudes absorbed in $s_k \sim \mathcal{N}(0, \alpha_s^{-1})$, with a gamma hyper-prior typically placed on α_s . We observed that the decomposition of amplitude (s_k) and direction (\mathbf{u}_k and \mathbf{v}_k) make the inference less sensitive to the prior setting and the initialization. It is possible to explicitly impose that the \mathbf{u}_k and \mathbf{v}_k are orthonormal [Hof09], but this has proven unnecessary (we have found in our experiments that using \mathbf{U} and \mathbf{V} to define a linear subspace is sufficient); the imposition of orthonormality comes at significant computational cost.

There are many similar approaches of the form in (5.3) without the presence of s_k and z_k , such as [DSG09], [LT07], probabilistic matrix factorization (PMF) [SM07], and Bayesian probabilistic matrix factorization (BPMF) [SM08]. Besides, Gaussian process latent variable models (GP-LVM) [LU09] non-linearly generalize this class of models. The special linear case of the GP-LVM is equivalent to the aforementioned models with one of the singular vector matrices marginalized. As mentioned in Chapter 1, a challenge is the need to estimate or define the dimensionality of the latent

space, i.e., the number of matrix components necessary for the reconstruction of the data matrix. A gamma process prior [HBC10] has been applied to encourage the shrinkage of the singular values in audio source separation, where those components with tiny singular values could be declared to be unused. The main distinction of the proposed model is that it provides an alternative method for inferring the aforementioned latent-space dimension, explicitly minimizing the l_0 norm of the pseudo singular values. As we will see in the next section, compared to shrinkage priors, we may save computation in a principled way.

5.3 Inference by Markov Chain Monte Carlo

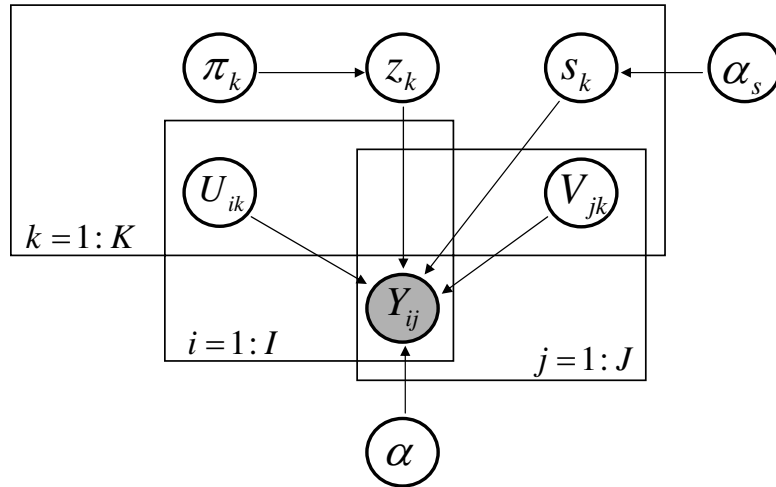


FIGURE 5.1: Graphical representation of the BSVD model. Refer to Figure 3.1 for additional information.

For the purpose of inference, we rewrite the BSVD model in a hierarchical form with the graphical representation shown in Figure 5.1. The pseudo singular values s_k are imposed to be non-negative with a truncation normal prior (denoted as \mathcal{TN})

for the sake of identifiability.

$$\begin{aligned}
Y_{ij} &\sim \mathcal{N}\left(\sum_{k=1}^K (s_k z_k) U_{ik} V_{jk}, \alpha^{-1}\right) \text{ for } (i, j) \in \Omega; \\
s_k &\sim \mathcal{TN}(0, \alpha_s^{-1}, s_k > 0), \text{ for } k = 1, \dots, K; \\
z_k &\sim \text{Bern}(\pi_k), \quad \pi_k \sim \text{Beta}(a/K, b(K-1)/K), \text{ for } k = 1, \dots, K; \\
U_{ik} &\sim \mathcal{N}\left(0, \frac{1}{I}\right), \text{ for } k = 1, \dots, K, \quad i = 1, \dots, I; \\
V_{jk} &\sim \mathcal{N}\left(0, \frac{1}{J}\right), \text{ for } k = 1, \dots, K, \quad j = 1, \dots, J, \\
\alpha &\sim \text{Gam}(c, d), \quad \alpha_s \sim \text{Gam}(e, f).
\end{aligned} \tag{5.3}$$

Since all the entries Y_{ij} are independent conditional on the model parameters s_k , z_k , U_{ik} , and V_{jk} , the marginalization of the missing entries is trivial. The data likelihood could be expressed as the likelihood of only the observed entries, i.e.,

$$p(\mathbf{Y} | \mathbf{s}, \mathbf{z}, \mathbf{U}, \mathbf{V}) = \prod_{(i,j) \in \Omega} \mathcal{N}\left(\sum_{k=1}^K (s_k z_k) U_{ik} V_{jk}, \alpha^{-1}\right).$$

The prediction on the missing entries in the data matrix \mathbf{Y} could be given by $\hat{Y}_{ij} = X_{ij} = \sum_{k=1}^K (s_k z_k) U_{ik} V_{jk}$ for $(i, j) \notin \Omega$, with the uncertainty on the prediction reflected by the inferred posteriors $p(\mathbf{s})$, $p(\mathbf{z})$, $p(\mathbf{U})$, and $p(\mathbf{V})$. The expected values for the prediction may be calculated by marginalizing those hidden parameters, i.e.,

$$\mathbb{E}[\hat{Y}_{ij}] = \int \int \int \int \sum_{k=1}^K (s_k z_k) U_{ik} V_{jk} p(\mathbf{s}) p(\mathbf{z}) p(\mathbf{U}) p(\mathbf{V}) d\mathbf{s} d\mathbf{z} d\mathbf{U} d\mathbf{V}, \text{ for } (i, j) \notin \Omega.$$

This marginalization is not analytical in general; however, it is straightforward to obtain not only the expectation but also the whole distribution of the predicted \hat{Y}_{ij} when those posteriors are represented by samples. Assume we have N samples of

parameters, i.e., $\{(s_k^{(n)}, z_k^{(n)}, U_{ik}^{(n)}, V_{jk}^{(n)})_{n=1}^N\}$, then each sample from the distribution of \hat{Y}_{ij} could be given by

$$\hat{Y}_{ij}^{(n)} = \sum_{k=1}^K (s_k^{(n)} z_k^{(n)}) U_{ik}^{(n)} V_{jk}^{(n)}, \quad \text{for } (i, j) \notin \Omega.$$

Knowing the uncertainty on prediction of missing entries is critical for selecting observed entries actively.

In Chapter 3 we have discussed two methods of Bayesian inference: the Markov chain Monte Carlo (MCMC) and the variational Bayesian (VB). In the context of classification problem considered in Chapter 3, the model has been implemented using variational Bayesian inference since we prefer that posteriors are of concise functional forms. However, here we choose to implement the proposed BSVD model using Gibbs sampling, the simplest sampling scheme of the MCMC inference with all the conditional density functions updated analytically. The updating is as follows:

1. Update s_k , for $k = 1, \dots, K$.

$$\text{Define } \mathbf{s}^{-k} = [s_1, \dots, s_{k-1}, s_{k+1}, \dots, s_K], \quad X_{ij}^{-k} = X_{ij} - \sum_{l \neq k}^K s_l z_l U_{li} V_{lj}.$$

$$(s_k | -) \sim \mathcal{TN}(\mu_{sk}, \alpha_{sk}^{-1}, s_k > 0) \text{ where}$$

$$\alpha_{sk} = \alpha_s + \alpha z_k^2 \sum_{(i,j) \in \Omega} U_{ik}^2 V_{jk}^2 \quad (5.4)$$

$$\mu_{sk} = \alpha_{sk}^{-1} \alpha z_k \sum_{(i,j) \in \Omega} X_{ij}^{-k} U_{ik} V_{jk} \quad (5.5)$$

2. Update z_k , for $k = 1, \dots, K$.

$$\frac{P(z_k = 1 | -)}{P(z_k = 0 | -)} = \frac{\pi_k}{1 - \pi_k} \exp\left\{-\frac{\alpha}{2} \sum_{(i,j) \in \Omega} (-2X_{ij}^{-k} s_k U_{ik} V_{jk} + s_k^2 U_{ik}^2 V_{jk}^2)\right\} \quad (5.6)$$

3. Update π_k , for $k = 1, \dots, K$.

$$(\pi_k | z_k) \sim \text{Beta}(a/K + z_k, b(K-1)/K + 1 - z_k) \quad (5.7)$$

4. Update U_{ik} , for $k = 1, \dots, K$, and $i = 1, \dots, I$.

$(U_{ik} | -) \sim \mathcal{N}(\mu_{U_{ik}}, \alpha_{U_{ik}}^{-1})$ where

$$\alpha_{U_{ik}} = I + \alpha s_k^2 z_k^2 \sum_{j:(i,j) \in \Omega} V_{jk}^2 \quad (5.8)$$

$$\mu_{U_{ik}} = \alpha_{U_{ik}}^{-1} \alpha s_k z_k \sum_{j:(i,j) \in \Omega} X_{ij}^{-k} V_{jk} \quad (5.9)$$

5. Update V_{jk} , for $k = 1, \dots, K$, and $j = 1, \dots, J$.

$(V_{jk} | -) \sim \mathcal{N}(\mu_{V_{jk}}, \alpha_{V_{jk}}^{-1})$ where

$$\alpha_{V_{jk}} = J + \alpha s_k^2 z_k^2 \sum_{i:(i,j) \in \Omega} U_{ik}^2 \quad (5.10)$$

$$\mu_{V_{jk}} = \alpha_{V_{jk}}^{-1} \alpha s_k z_k \sum_{i:(i,j) \in \Omega} X_{ij}^{-k} U_{ik} \quad (5.11)$$

6. Update α

$$(\alpha | -) \sim \text{Gam} \left(c + \frac{m}{2}, d + \frac{1}{2} \sum_{(i,j) \in \Omega} \left(X_{ij} - \sum_{k=1}^K s_k z_k U_{ik} V_{jk} \right)^2 \right). \quad (5.12)$$

7. Update α_s

$$(\alpha_s | -) \sim \text{Gam} \left(e + \frac{K}{2}, f + \frac{1}{2} \sum_{k=1}^K s_k^2 \right). \quad (5.13)$$

Note that when any z_k is sampled to be zero in some MCMC iteration, the conditional posteriors of s_k , U_{ik} , and V_{kj} reduce to their priors. That means we do not need to

update those conditional posteriors for the components with $z_k = 0$. Compared with methods [HBC10] with shrinkage non-parametric priors on singular values, we save a lot of computation on those unused components in a principled way by introducing the binary selecting variables.

5.4 Model Generalizations

We generalize the basic BSVD model in two directions: i) to make use of available information more wisely and more efficiently; ii) to extend the model to fit into data matrices with integer counts as entries. Although we are making these generalizations based on the proposed BSVD model, they could be applied on other matrix decomposition models without difficulty.

5.4.1 Auxiliary information

In many collaborative-filtering (CF) problems we may have feature vectors $\mathbf{r}_i \in \mathbb{R}^{J_r}$, for $i \in \{1, \dots, I\}$, with \mathbf{r}_i representing observed covariates associated with the i th row of \mathbf{Y} . Similarly, we may have $\mathbf{c}_j \in \mathbb{R}^{J_c}$ for $j \in \{1, \dots, J\}$, representing features of the J columns. For example, other than the rating matrix we may have additional information on the age, occupation and gender of users, and the genre, year and director of movies in the movie ratings problem. Even if these additional information is not available, we can always know which ratings are missing in the data matrix. This is also one kind of auxiliary information. Although existing CF methods always assume missing at random (MAR), it is commonly believed that the MAR assumption is not true in CF problems. Usually people tend to watch and rate those movies they might like; therefore, the missingness of ratings is very likely to be related to the values. We also have made the MAR assumption for the basic BSVD model, and we expect to enhance our model by using the information on missingness. All these auxiliary information could be casted in the model as regression functions.

Specifically, consider the following model

$$\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{R}\mathbf{G}_R^T + \mathbf{G}_C\mathbf{C}^T + \mathbf{E} \quad (5.14)$$

where $\mathbf{S} = \text{Diag}(\mathbf{z} \circ \mathbf{s})$, $\mathbf{R} \in \mathfrak{R}^{I \times J_r}$ and $\mathbf{C} \in \mathfrak{R}^{J \times J_c}$ are known features for rows and columns, respectively, and $\mathbf{G}_R \in \mathfrak{R}^{J \times J_r}$ and $\mathbf{G}_C \in \mathfrak{R}^{I \times J_c}$ are regression matrices to be inferred. The data matrix \mathbf{Y} is explained jointly by exchangeable bilinear random effects, nonexchangeable (with constraints given by features) linear random effects and errors. In this sense, the model is related to [YLZG09], where the exchangeable random effects model is assumed to be a nonparametric Gaussian process and thus the error term is absorbed.

5.4.2 Active learning

There are problems for which one may ask questions relative to \mathbf{Y} , to gather information that may enhance our ability to perform inference. For example, assume that \mathbf{Y} represents a matrix of (highly incomplete) movie ratings, where one axis corresponds to people and the other to movies. Based upon the proposed Bayesian approach (as well as other previous related Bayesian methods [LT07, SM08, YLZG09]) one may infer a posterior density function on the components of \mathbf{Y} . One may then ask selected individuals to watch selected movies (*e.g.*, at no cost, or with some compensation), with movie ratings requested. By selecting these individuals and movies appropriately, one may improve the ability to perform inference at the many other points in \mathbf{Y} for which data are unavailable. One can imagine many related matrix-type problems for which this form of active learning may be employed, for example for rating music or other products.

Another important application of active learning could be problems with large-scale data matrices where many of observed entries may not be informative for the inference of unknowns. The computational cost could be highly reduced by selecting

a relative small number of informative entries to acquire for each step.

Here we *begin* to explore this problem, recognizing that there is much work to be done in the future. The most naive approach would simply determine which components of \mathbf{Y} have greatest posterior uncertainty, and seek ratings on those elements with greatest uncertainty, and then refine the estimates of \mathbf{Y} based upon the augmented available data. Since our analysis is Bayesian, we manifest “error bars” on the inferred matrix values, and we demonstrate how these may be used in an active-learning setting to selectively acquire matrix values and often significantly improve learning performance.

This is clearly suboptimal, as many of these uncertain elements of \mathbf{Y} may be highly correlated, and thus the information gain of newly acquired entries could be limited after we have already seen some of the correlated ones. Therefore the objective is to select those elements of \mathbf{Y} that are uncertain *and* uncorrelated. Related research has been pursued using submodularity methods on a different class of problems [KG07] (and submodularity may be a fruitful research direction for this problem).

5.4.3 Probit link function

In the above discussion we assumed that the (partially) observable data matrix was real. In many applications the observed data may be integers. For example, in the movie-rating case, the observed data are often integer ratings, that are often just regarded as real numbers. It is relatively straightforward to consider integer data within a Bayesian setting, by introducing a link function [AC93]. Here we employ a probit link function, and assume L integer values (*e.g.*, $L = 2$ for binary data). In a probit-regression model [MN89] we introduce $L - 1$ real numbers $\phi_1 < \phi_2 < \dots < \phi_{L-1}$. The matrix decomposition is still operated on the latent continuous \mathbf{X} matrix defined earlier; however, the noisy matrix $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{E}$ is also latent with $\tilde{\mathbf{X}}$ linked to

\mathbf{Y} via the probit link function. The (i, j) th element is mapped to 1 if $\tilde{X}(i, j) \leq \phi_1$, to value L if $\tilde{X}(i, j) > \phi_{L-1}$, and to l for $1 < l < L$ if $\phi_l < \tilde{X}(i, j) \leq \phi_{l+1}$. For $L = 2$ we typically set $\phi_1 = 0$, and for $L > 2$ a prior is typically set on $\{\phi_l\}_{l=2, L-1}$, with $\phi_1 = 0$; in our work we consider a broad Gaussian prior for $\{\phi_l\}_{l=2, L-1}$. The Gaussian noise precision α is set to be 1 for identifiability. Instead of a common set of thresholds, a unique set $\{\phi_{il}\}_{l=2, L-1}$ may be considered for individual user i as in [DSG09]; however, the number of unknowns will increase dramatically, and from our experience this setting may leave too much flexibility so that the identifiability becomes an issue.

As discussed below when presenting results, for the movie-rating problem, treating the data as integer and using a probit link was found to yield performance gains, but these gains are probably not worth the additional computational cost, *vis-a-vis* simply assuming the observed data are real. However, there are other problems for which the probit model is essential, and we introduce data for such a problem in Section 5.5. Specifically, we consider two data matrices from Major League Baseball (MLB) in the United States. One axis of the matrices corresponds to baseball hitters, and the other to pitchers. The (i, j) element in the data matrix \mathbf{Y}_p records the number of events that baseball hitter i is pitched to by pitcher j . Among the $\mathbf{Y}_p(i, j)$ events, hitter i gets a hit for $\mathbf{Y}_h(i, j)$ times when being pitched to by pitcher j . Both \mathbf{Y}_p and \mathbf{Y}_h are strong functions of (i, j) , with most entries being zeroes. The objective could be to estimate the probability of a hit for all (i, j) , which would be useful for predicting future performance of baseball players. Via collaborative filtering, and the inference of relationships between different hitters and pitchers, the probability of hitter i being successful against pitcher j may be estimated, even when $\mathbf{Y}_p(i, j)$ is small or zero. The key difference from conventional probit model is that here the observed data are the number of total binary trials and the number of successful trials among them instead of one binary outcome for each (i, j) . To

handle such problems with $Y_h(i, j)$ successful trials among total $Y_p(i, j)$ binary trials, we employ the probit model as follows.

$$Y_h(i, j) = \sum_{t=1}^{Y_p(i, j)} Q_t(i, j), \text{ where } Q_t(i, j) = \begin{cases} 0, & \tilde{X}_t(i, j) < 0 \\ 1, & \tilde{X}_t(i, j) > 0 \end{cases};$$

$$\tilde{X}_t(i, j) \sim \mathcal{N}\left(\sum_{k=1}^K (s_k z_k) U_{ik} V_{jk}, 1\right), \text{ for } t = 1, \dots, Y_p(i, j),$$

where $Q_t(i, j)$ is the latent binary outcome of the t th trial for (i, j) .

The data from this problem will be made available to the community, and introduces a new class of problems that may be solved with collaborative filtering (we assembled data for all MLB batter-hitter events from 1954-2008). Detailed discussion is in Section 5.5.

5.5 Example Results

In this section we first implement the basic BSVD model on synthetic matrices and demonstrate the effectiveness of the active learning. After that we test the proposed BSVD model on a widely used benchmark data set: 1M MovieLens movie-rating data set with auxiliary information. At last, a new data set is introduced to the community, and interesting results are provided by the generalized model.

5.5.1 Parameter settings

Noninformative Gamma priors are put on the precision of noise and the precision of pseudo singular values, i.e. hyper-parameters $c = d = e = f = 10^{-6}$ throughout all the experiments. For the sake of computational practice, instead of setting a ridiculously large value for the number of candidate factors K , we suggest to first set a reasonably large value for K and increase it only if candidates are fully occupied, which may indicate insufficiency of candidate factors. In all the following experiments

K is fixed to be 50 and candidates are never fully occupied.

Low-rank belief is imposed through a and b , the hyper-parameters of prior for π_k . We first examine the influence of different settings for a and b on synthetic examples. We observed that for middle to large-scale problems the model inference is not sensitive to the actual values of a and b . In all the experiments conducted on real data sets, we set $a = b = 1$ (no tuning of hyperparameters). This is considered to be an important advantage over other algorithms which usually rely on cross-validation to tune parameter settings, especially the dimensionality of the latent space.

5.5.2 Synthetic matrices

The purpose of simulation analysis is to examine the performance of the proposed BSVD model given a variety of matrix sizes I , ranks r and numbers of observed entries m . A data matrix $\mathbf{Y} \in \mathfrak{R}^{I \times I}$ of rank r is generated by sampling two $I \times r$ matrices \mathbf{Y}_L and \mathbf{Y}_R with i.i.d. Gaussian entries and setting $\mathbf{Y} = \mathbf{Y}_L \mathbf{Y}_R^T$. For any given matrix \mathbf{Y} , we uniformly sample m entries at random to be observed and other entries are missing. The BSVD model is learned from the m observed entries, and the recovered matrix $\hat{\mathbf{Y}}$ is given by the posterior mean. We define the relative reconstruction error $\varepsilon = \|\hat{\mathbf{Y}} - \mathbf{Y}\|_F / \|\mathbf{Y}\|_F$ and declare \mathbf{Y} to be recovered successfully if $\varepsilon < 10^{-3}$. The whole procedure is repeated 50 times and the empirical recovery rate is calculated for each $(I; m; r)$ triple.

The recovery rate for $I = 50$ is shown in Figure 5.2 (b) as a joint function of d_r/m and m/I^2 , where $d_r = r(2I - r)$ denotes the degrees of freedom for an $I \times I$ matrix of rank r . The horizontal axis indicates the fraction of entries that are observed, therefore it should be in the range of $(0, 1]$ and the recovery rate should tend to increase when we go right. The vertical axis reflects the relationship between the degrees of freedom and the observed entries. This axis value could be larger than

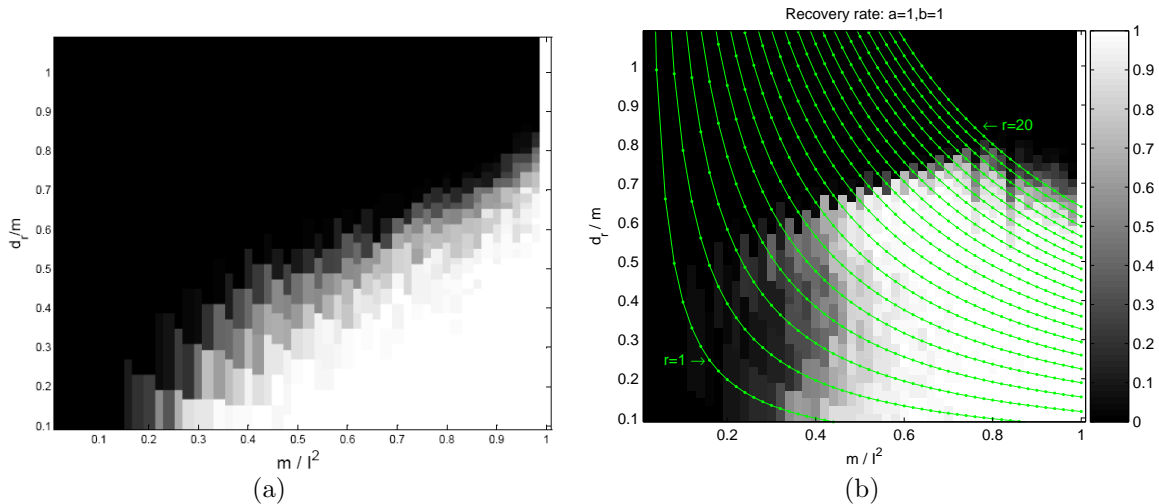


FIGURE 5.2: Empirical recovery rate of full matrices over 50 random trials for each rank r and number of observed entries m (matrices size $I = 50$). The degrees of freedom $d_r = r(2I - r)$. A matrix \mathbf{Y} is declared to be recovered if the reconstructed matrix $\hat{\mathbf{Y}}$ satisfies $\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F / \|\mathbf{Y}\|_F < 10^{-3}$. Results are for (a) [CR09] and (b) BSVD ($a = b = 1$). The green lines are equal-rank trajectories with the rank increasing monotonously from the lower left to the upper right.

one; however in that case the number of unknowns is larger than the number of equations and thus we may have infinite number of solutions and exact recovery is not possible. As a result, we consider the vertical axis in the range of $(0, 1]$, and the recovery rate should tend to increase when we go down. To make the tendency more clear, we plot equal-rank trajectories in green in Figure 5.2 (b). Along each green line from upper left to lower right, the rank of matrices is fixed while the number of measurements m increases. Figure 5.2 (a) is cited from [CR09], where the solution is given by minimizing the nuclear norm subject to the constraint of observed entries. It is clear that the proposed BSVD model pushes the white area (high recovery rate) up to regions of higher d_r/m and lower observation rate m/I^2 when the rank of matrices is around 5-15.

To examine the influence of different settings for a and b , we present the results for $a = 1, b = 10$ and $a = 10, b = 1$ in Figure 5.3. As shown in Figure 5.3 the $a = 1, b =$

10 setting is good for matrices of rank less than 7; while the $a = 10, b = 1$ setting encourages better results for matrices of rank more than 6. Recall that the prior expected number of non-zero z_k 's (pseudo rank) is given by $aK/(a+b(K-1)) \approx a/b$ when K is very large. Our priors on the rank of matrices are actually around 1, 0.1, and 10 for the settings in Figure 5.2 (b), Figure 5.3 (a) and (b), respectively. Although these priors have impact on the model learning, the posterior beliefs are not necessary the same as the prior since we update our prior beliefs after seeing a data matrix. The results presented here are sensitive to the choice of a and b as we are working on small-scale matrices for the sake of comparison with [CR09]. However, for large-scale problems, all these three settings may convey roughly the same prior information that the pseudo rank of the matrix should be small. In the analysis for real data matrices of much larger scale, we fix $a = b = 1$.

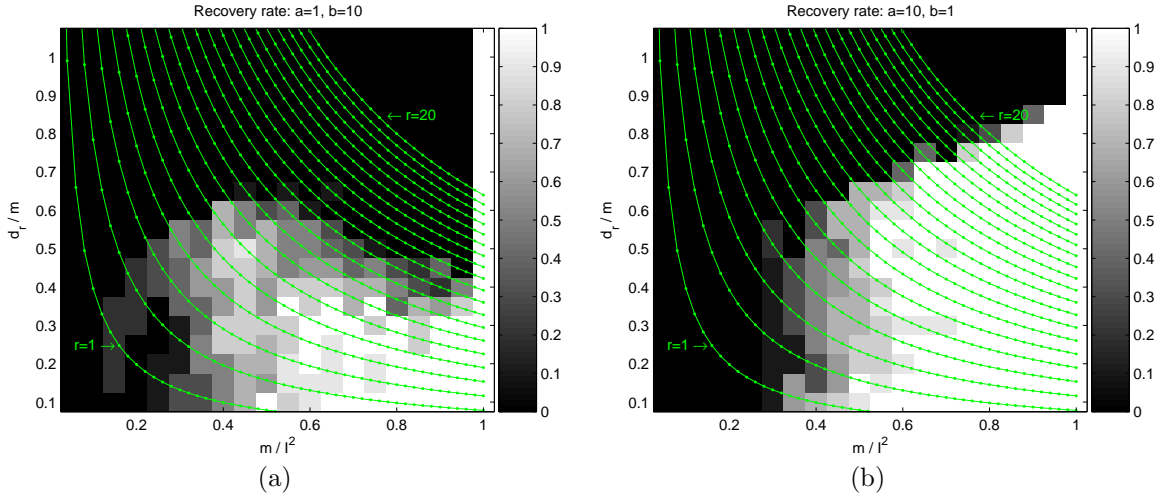


FIGURE 5.3: Empirical recovery rate of full matrices given by the BSVD when (a) $a = 1, b = 10$ and (b) $a = 10, b = 1$. Refer to Figure 5.2 for additional information.

An important advantage of Bayesian approaches is that we not only give the reconstruction itself and also provide how certain we are about the reconstruction. This provides us the opportunity to acquire entries actively. After generating matrices $\mathbf{Y} \in \mathbb{R}^{50 \times 50}$ of rank r as before, we first randomly sample 50 entries to be

observed for model learning, and then simply select 50 unseen entries with the highest standard deviation in prediction to add into observed data in the next step. This procedure is repeated until all the entries are observed. After the second step, the last MCMC sample in the previous step is used to initialize the current model. As a comparison, we execute the same serial of experiments except that we augment the observed data by acquiring 50 entries randomly in each step. We consider that the rank r varies from 1 to 10, and show some representative results for $r = 1, 5, 10$ in Figure 5.4. For each value of r , we simulate 10 matrices and conduct 10 independent data acquiring procedures. The mean and standard deviation of reconstruction errors are reported. From Figure 5.4, more entries are required to reconstruct matrices of higher rank in general; by active learning, we can either achieve a lower reconstruction error with the same amount of entries or use less data to achieve the same performance.

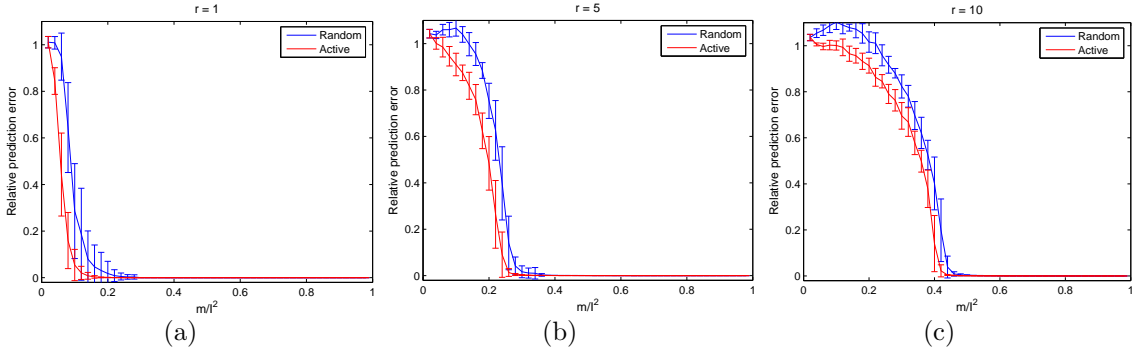


FIGURE 5.4: Sequentially acquire entries from noise-free matrices ($I = 50$) of rank (a) $r = 1$, (b) $r = 5$, and (c) $r = 10$. Blue corresponds to uniformly sampling entries at random; red represents selecting entries actively. Error bars reflect the standard deviation of reconstructed errors over 10 independent trials.

We also examine the situations that matrices are not exactly low-rank, i.e., noises are introduced to low-rank matrices. Two signal-to-noise ratios are considered (20dB and 10dB) in Figure 5.5. Similar observation could be made as for the noise-free cases in Figure 5.4 except that perfect reconstructions are no longer possible because

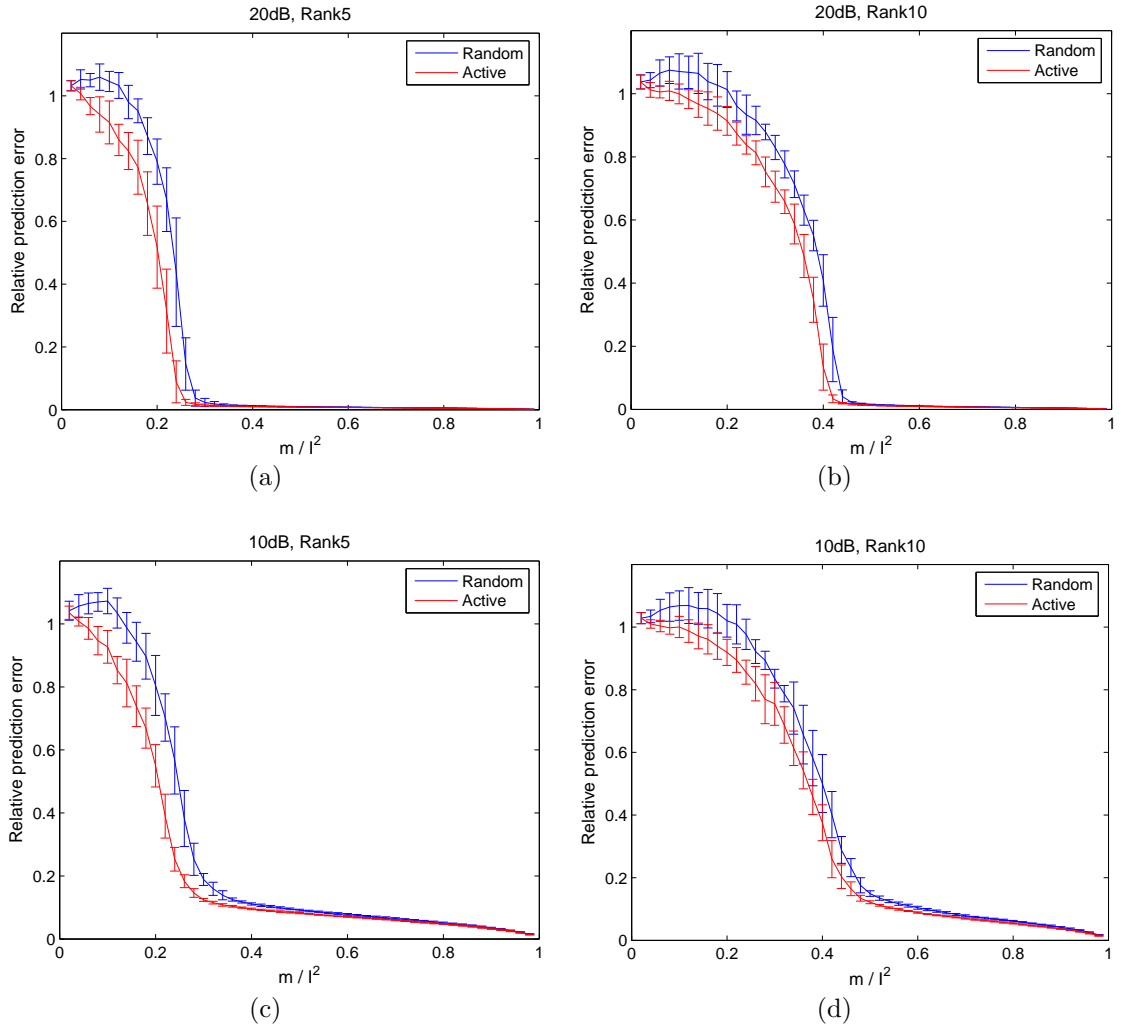


FIGURE 5.5: Sequentially acquire entries from noisy matrices ($I = 50$). (a) $SNR = 20dB$, $r = 5$, (b) $SNR = 20dB$, $r = 10$, (c) $SNR = 10dB$, $r = 5$, and (d) $SNR = 10dB$, $r = 10$.

of the present of noises.

5.5.3 Movie ratings

We validate the proposed algorithm on 1M MovieLens movie-rating data set (available at <http://www.grouplens.org/node/73>), which is a widely used benchmark [Mar04, RS05, DeC06, LU09]. The base data set of 1M MovieLens contains about 1 million ordinal ratings for 3,952 movies by 6,040 users, with a 5-star scale. About 95.7% of entries in this data matrix are missing. Besides the base rating matrix, auxiliary

Table 5.1: Auxiliary features in 1M MovieLens data set.

| Movie genre | User gender | User age | User occupation |
|------------------------|-------------|----------|------------------------|
| Action | F | Under 18 | Academic/Educator |
| Adventure | M | 18-24 | Artist |
| Animation | | 25-34 | Clerical/Admin |
| Children's | | 35-44 | College/Grad student |
| Comedy | | 45-49 | Customer service |
| Crime | | 50-55 | Doctor/Health care |
| Documentary | | 56+ | Executive/Managerial |
| Drama | | | Farmer |
| Fantasy | | | Homemaker |
| Film-Noir | | | K-12 student |
| Horror | | | Lawyer |
| Musical | | | Programmer |
| Mystery | | | Retired |
| Romance | | | Sales/Marketing |
| Sci-Fi | | | Scientist |
| Thriller | | | Self-employed |
| War | | | Technician/Engineer |
| Western | | | Tradesman/Craftsman |
| Other or not specified | | | Unemployed |
| | | | Writer |
| | | | Other or not specified |

features describing the movies and the users are available as shown in Table 5.1. There are total 18 possible genres, and any one movie may be a combination of several of them. Those movies without a genre definition go to the category “Other or not specified”. Any one user is described by his/her gender (F/M), age (7 possible ranges) and occupation (21 possible values). A SVD decomposition is performed for the 0-1 missingness matrix (0-missing; 1-observed), and first 10 left and right singular vectors are also used as the features for movies and users, respectively.

For a fair comparison, we conducted the same experiments as defined by Marlin [Mar04] and followed by [RS05, DeC06, LU09]. Three random trials are performed, with 5000 users randomly selected for each trial. For each user one randomly selected rating is hold for test. As a result, for each trial about 830000 ratings are used for model training and 5000 ratings for evaluation. Exactly the same three random par-

titions of training and testing sets as in [Mar04] are considered, and the mean with standard deviation of the prediction errors for these three trials is reported in Table 5.5.3. In [LU09] GP-LVM was demonstrated to yield superior performance relative to many of the algorithms in the literature, and therefore for brevity here we only include the GP-LVM [LU09] results here. As the linear special case of the GP-LVM is equivalent to the SVD decomposition models with a preset rank, the comparison between the proposed BSVD and the linear GP-LVM could be interesting. By comparing the BSVD to the generalized GP-LVM, we may get some intuition about whether the nonlinear assumption in the GP-LVM helps on this data set.

Both the root mean squared error (RMSE) and normalized mean absolute error (NMAE) are used as performance measures. The RMSE and the MAE are defined as follows:

$$RMSE = \sqrt{\frac{1}{|\Omega_{test}|} \sum_{(i,j) \in \Omega_{test}} (Y(\hat{i}, j) - Y(i, j))^2},$$

$$MAE = \frac{1}{|\Omega_{test}|} \sum_{(i,j) \in \Omega_{test}} |Y(\hat{i}, j) - Y(i, j)|.$$

The NMAE is defined to be the normalized MAE, where the normalizing constant is defined to be the expected value of the MAE when both the observed and the predicted rating values are uniformly distributed [Mar04]. Therefore, an NMAE value of one means predicting the ratings by random. The normalizing factor depends on the range of the ratings. On a scale from one to five as in the 1M MovieLens data set, this normalizing constant equals to 1.6.

Auxiliary features have been used to improve the accuracy of predictions in different ways. For example, in [LU09] the information on the movie genre was included into an extra factor of kernel matrix in Gaussian process; while in [YLZG09] the missingness was used as side information. Here we consider four kinds of auxiliary

Table 5.2: Prediction errors on 1M MovieLens data set. Mean and standard deviation over the three partitions are reported. For the proposed BSVD model, the codes “0000”, “0011”, “1100” and “1111” correspond to the basic model as in (5.3), the models with user auxiliary information, movie auxiliary information, and all available auxiliary information, respectively. Results for the GP-LVM are cited from [LU09], with the latent dimensionality yielding these best results indicated.

| 1M MovieLens | | | |
|--------------|----------|---------------------------|---------------------------|
| Methods | Settings | NMAE | RMSE |
| GP-LVM | linear | 0.4052 ± 0.0011 (11D) | 0.8791 ± 0.0080 (14D) |
| | RBF | 0.4026 ± 0.0020 (10D) | 0.8801 ± 0.0082 (12D) |
| BSVD | 0000 | 0.3942 ± 0.0026 | 0.8614 ± 0.0088 |
| | 0011 | 0.3933 ± 0.0037 | 0.8598 ± 0.0082 |
| | 1100 | 0.3914 ± 0.0056 | 0.8588 ± 0.0067 |
| | 1111 | 0.3922 ± 0.0052 | 0.8584 ± 0.0070 |

information: the movie genre; movie missingness features; user meta data which includes age, gender and occupation; and the user missingness features. The usage of these auxiliary information is encoded in a binary vector. For example, “1100” means the movie genre and movie missingness features are incorporated into the basic model. As shown in Table 5.1, a 19-dimensional binary vector can be used to define the genre for each movie, and each user corresponds to a 29-dimensional binary vector including information on age, gender and occupation. Similar to [YLZG09], missingness feature vectors were obtained by factorizing the binary matrix indicating which ratings are observed. These auxiliary features were included using the framework in (5.14). The way to handle the metadata and missingness features are not restricted to the model proposed in the paper and could be applied to other bayesian models to get improved performance.

According to Figure 3 in [LU09], with an extra kernel function for the movie genre information, the GP-LVM achieves the best NMAE (0.4042) for a 14D model and the best RMSE (0.8755) for a 12D model. As can be seen in Table 5.5.3, the basic model (0000, without auxiliary information) of our approach already gives better

performance than the GP-LVM [LU09] with both linear and RBF kernels, and even the GP-LVM with the movie genre information. Further improvement is achieved by adding the user auxiliary information (0011), the movie auxiliary information (1100), or all the available auxiliary information (1111).

Importantly, the dimensionality of the latent space has to be tuned to yield best results for the GP-LVM and most other similar models; however, we do not need to try different values for the latent dimensionality, or perform cross-validation to deal with this model-selection issue. The algorithm infers the dimensionality of the latent space by itself. For example, the learned dimensionality is respectively 23, 27, and 26 for the basic BSVD model on the three partitions; for the model with all the available auxiliary information (1111), the learned dimensionality is respectively 29, 28, and 29. It is noticed that in [LU09] the latent dimensionality is considered up to 15. We appreciate the awareness of possible over-fitting issues incurred by setting a much higher dimensionality; however, for small training sets even 10D may be over fitting, while for large and complex training sets 15D may be still far from adequate. Moreover, even for training sets with comparable ratings, the appropriate dimensionality of latent space may be different. Therefore, the ability of inferring the dimensionality of latent space is critical for algorithms.

To make this point more clear, we conduct a further experiment using the basic BSVD model, where we randomly separate the ratings for training and testing, with five random trials for each training size as in [LU09]. The averaged RMSE and NMAE are shown in Figure 5.6, together with representative GP-LVM results cited from Figure 2 in [LU09]. Figure 5.6 shows that the proposed BSVD model outperforms the GP-LVM with any dimensionality of latent space from 2D to 10D, for all the training sizes under consideration. We also observed that the preset dimensionality highly influences the performance of the GP-LVM. It seems that the GP-LVM with higher dimensionality (e.g., 6D or 10D) suffers from over-fitting when training size is

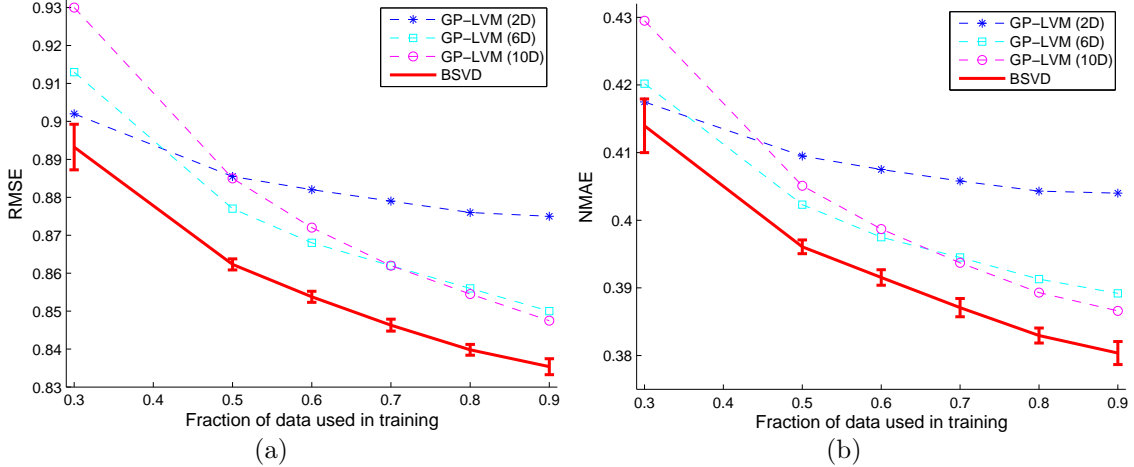


FIGURE 5.6: 1M MovieLens: (a) RMSE and (b) NMAE as a function of the fraction of data used for training (five random selections of training data for each training size). The results of the GP-LVM are cited from Figure 2 in [LU09]. To make the figures legible, we present only the performance means for the GP-LVM with 2D, 6D and 10D latent space, which are representative. The performance means and standard deviations are reported for the proposed BSVD model.

small; however, the model with lower dimensionality (e.g., 2D) seems not capable of explaining well training data of large size. This discloses the importance of inferring the dimensionality in the light of training data. The inferred rank (dimensionality of latent space) by the BSVD is included in Figure 5.7. It can be seen that for this data set the number of necessary latent factors increases nearly linearly as the training size increases. Even with a fixed training size, the latent dimensionality could also be different from trial to trial.

We also tested the proposed method on the 1 million MovieLens data set using the probit link function discussed in Section 5.4.3, and found that the inferred thresholds ϕ_i were very close to half-star increments that are widely used to convert real matrix values to the integer-star values [DeC06]. Therefore, for this problem we did not find the additional computational cost required to learn the thresholds worthwhile. However, in Section 5.5.4 the form of the probit link function is essential.

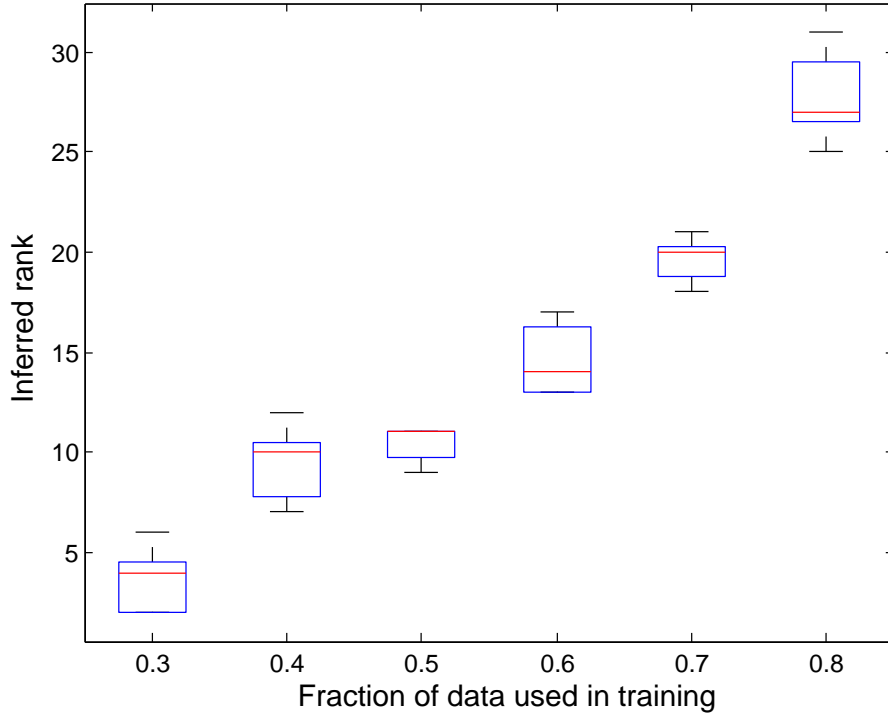


FIGURE 5.7: 1M MovieLens: Inferred rank by the proposed BSVD model as a function of data used for training. Each box plot represents the distribution of the rank inferred in five trials for one training size. Detailed explanation for box plots is in the caption of Figure 3.8.

5.5.4 MLB data from 1954-2008

At www.retrosheet.org one may acquire data on every batter-pitcher event that occurred in Major League Baseball (MLB), from 1954-2008. This may be viewed as a matrix, corresponding to all $I = 7693$ MLB hitters who have had an at-bat during that period, and all $J = 4561$ pitchers who have pitched. For hitter $i \in \{1, \dots, I\}$ and pitcher $j \in \{1, \dots, J\}$, there are $Y_p(i, j)$ events at which these two players have competed, and a one is recorded in $Y_h(i, j)$ if the hitter got a hit, and a zero is recorded if not (there are $Y_p(i, j)$ binary values). For these data, 96.37% of the matrix has $Y_p(i, j) = 0$, meaning hitter i and pitcher j never met and 98.83% of the data has $Y_p(i, j) \leq 5$. Since most hitter-pitcher pairs met very few times, it is not

Table 5.3: MLB data log-likelihood for the year of 2008, evaluated for pitchers and batters appearing in training sets: 2007, 2003 to 2007, and 1954 to 2007, respectively.

| Methods | 2007 | 2003-2007 | 1954-2007 |
|--------------|-----------------------|-----------------------|-----------------------|
| Empirical | -2.1355×10^6 | -2.0209×10^6 | -1.9794×10^6 |
| Batter Avg. | -1.5843×10^5 | -1.2371×10^5 | -1.2240×10^5 |
| Pitcher Avg. | -8.6514×10^4 | -1.0130×10^5 | -1.0144×10^5 |
| BSVD | -8.3900×10^4 | -8.8518×10^4 | -8.8619×10^4 |

reliable to predict their future performance based merely on the historic records. Fortunately, for a given hitter-pitcher pair (i, j) we actually have more information than $Y_p(i, j)$ and $Y_h(i, j)$. For example, other than pitcher j , hitter i may have been also pitched by pitchers j_1, j_2, \dots ; and other than hitter i , pitcher j may have also pitched hitters i_1, i_2, \dots . These records may provide indirect information on the meet of hitter i and pitcher j . Our objective is to infer the latent *probability* that hitter i will be successful against pitcher j , using collaborative filtering. The probability of a hit for (i, j) is quantified via the probit link function.

To our knowledge, this is the first time these data have been analyzed using any collaborative-filtering method, and therefore future studies will help define the best way to display and compare results. Here we simply provide example results that seem to indicate the model is working properly, and also to show the types of interesting questions the analysis allows one to examine.

Since the result of an event that a hitter meets with a pitcher is a random outcome of a bernoulli trial, it is not appropriate to evaluate algorithms with prediction errors as we did in Section 5.5.3. As a means of quantifying the performance of the model, in Table 5.3 we present the log-likelihood of the observed 2008 data, based on models learned using data from 2007, 2003 through 2007, and 1954 through 2007, respectively. Only the likelihood for players appearing in these training sets is evaluated. Among 983 batters and 651 pitchers playing in 2008, respectively 759 and 528 played in the previous five years, and respectively 686 and 479 played in 2007.

In addition to examining results from the proposed Bayesian model, we also consider typical methods based on the cumulative average probability of success for all batter-pitcher pairs (“empirical”) in the training set, as well as average performance across rows (“batter average”) and across columns (“pitcher average”). For the “empirical” method, pitcher average probabilities are used for batter-pitcher pairs not appearing in training sets. Unsurprisingly, the proposed BSVD model outperforms the baselines. As the data matrices are fairly sparse, the empirical probability of hitting for (i, j) is highly unreliable for prediction of future events; while the average across rows or columns is more reliable.

We perform another experiment on the data from 2003 to 2007 to evaluate the model performance, where the ten batter-pitcher pairs with the largest $Y_p(i, j)$ (ranging from 62 to 70) are held out for testing, and the hitting probabilities are inferred from the remaining data. The ten predicted probabilities range from 0.24 to 0.29 with standard deviations around 0.014; the empirical (observed) hitting rates of the ten held-out cases lie in a range of 0.0128 to 0.0862 about the predicted probabilities. This experiment shows that we can predict random events reasonably well using the predicted probabilities given by the proposed model.

Bayesian methods give a measure of confidence in a prediction. In Figure 5.8 we plot the standard deviation of the latent probability that hitter i will be successful against pitcher j given all the available data (1954-2008). Specifically, we consider the standard deviation of each hitter’s success probability, averaged across all possible pitchers he may face. Batters are sorted in ascending order of the total number of events each has participated in. The total number of events for a given batter ranges from 1 to 16492 with a median 143 (half of the batters have less than 143 at-bats). The results indicate, as expected, that the standard deviation decreases significantly (from 0.25 to 0.05) with the number of at bats for batters. Similar observations are found for pitchers.

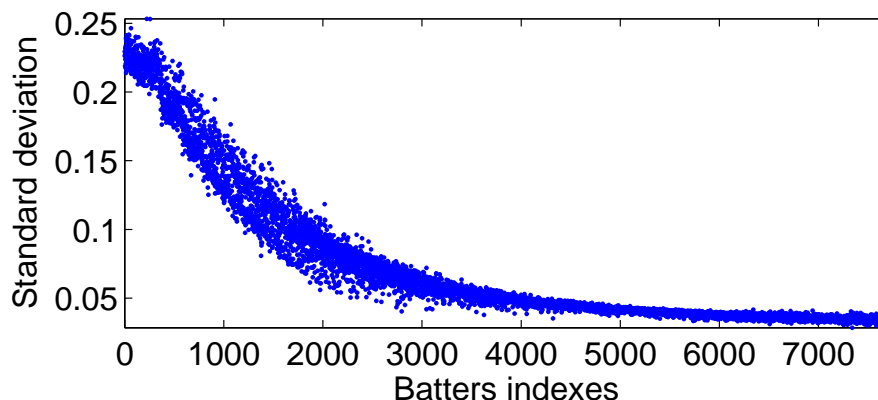


FIGURE 5.8: Average standard deviation of predicted probabilities of batters being successful against pitchers (1954-2008). Batters are sorted in ascending order of the total number of events each has participated in. The total number of events for a given batter ranges from 1 to 16492 with a median 143 (half of the batters have less than 143 at-bats).

Table 5.4: Prediction on probabilities of top batters being successful against top pitchers (means and standard deviations). Top batters: batting in at least the median number of events among all batters in a given year, and with the highest empirical hitting rates; top pitchers: pitching in at least the median number of events among all pitchers in a given year, and with the lowest empirical hitting rates.

| | | 1954 Pitchers | | | 2008 Pitchers | | |
|-----------------|-------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| | | D. Mossi | B. Turley | R. Narleski | C. Marmol | G. Balfour | B. Morrow |
| 1954 Batters | D. Mueller | 0.2829 ±0.0166 | 0.2340 ±0.0171 | 0.2543 ±0.0176 | 0.1472 ±0.0192 | 0.1755 ±0.0217 | 0.1627 ±0.0211 |
| | S. Burgess | 0.2619 ±0.0147 | 0.2277 ±0.0131 | 0.2391 ±0.0152 | 0.1338 ±0.0163 | 0.1550 ±0.0195 | 0.1502 ±0.0154 |
| | B. Skowron | 0.2657 ±0.0135 | 0.2141 ±0.0133 | 0.2339 ±0.0144 | 0.1254 ±0.0153 | 0.1530 ±0.0200 | 0.1407 ±0.0193 |
| 2008 Batters | C. Zambrano | 0.2652 ±0.0217 | 0.2215 ±0.0225 | 0.2384 ±0.0219 | 0.1313 ±0.0235 | 0.1566 ±0.0276 | 0.1475 ±0.0267 |
| | P. Sandoval | 0.3570 ±0.0355 | 0.3316 ±0.0400 | 0.3402 ±0.0386 | 0.2612 ±0.0529 | 0.2792 ±0.0500 | 0.2740 ±0.0522 |
| | R. Furcal | 0.2971 ±0.0140 | 0.2599 ±0.0141 | 0.2748 ±0.0144 | 0.1721 ±0.0158 | 0.1948 ±0.0187 | 0.1876 ±0.0189 |

As an example of a fun/interesting question one may ask, in Table 5.4 we consider the probability of success for the top three hitters from 1954 and 2008, with these hitters facing the top three pitchers from 1954 and 2008. This allows us to provide a principled answer to how players who played 54 years apart may have performed

if they were able to compete in their (presumed) primes. While there is of course no “truth” for this example, note that hitters from 1954 and 2008 do much better on average against top 1954 pitchers than against top 2008 pitchers.

5.6 Summary

We propose a new matrix-completion approach based on singular value decomposition (SVD) from a Bayesian perspective. A low-rank belief is imposed statistically via a Beta-Bernoulli prior on the binary variables selecting candidate latent factors. The model is unlikely prone to over-fitting and model selection issues are avoided since the number of necessary factors is inferred automatically in the light of data. The basic model is generalized to incorporate auxiliary information [YLZG09, ABEV09], acquire new data actively, and fit into data matrices with integer counts as entries.

We examine the algorithm on simulated matrices of various ranks, with different numbers of observed entries. Performance improvement is observed by acquire new data actively for both noise-free and noisy cases. We then demonstrate state-of-the-art performance on a typical movie-rating data set and disclose the advantage of inferring the latent dimension automatically. Auxiliary data including missingness information is incorporated to further improve the performance. Finally, a new data set is presented and analyzed using the proposed model generalized with a probit link function, where outcomes from binomial process are observed as data matrices while the real probability matrix is latent.

Conclusions and Future Work

6.1 Conclusions and Discussions

As we have mentioned, the presence of incomplete data is an unideal setting for most learning approaches, while it occurs in many applications due to various reasons; therefore, handling incomplete data is an important and challenging task for the machine learning community. In this dissertation we have addressed two kinds of problems in the presence of incomplete data in the Bayesian framework. One is classification with features partially missing, and the other is low-rank matrix completion from a portion of the matrix entries. By assuming appropriate Bayesian models and priors, we have integrated out the missing values analytically during the model inference. Model selection and over-fitting issues have been circumvented by considering the uncertainty on both parameters and model structures. We also have made use of some desirable aspects of Bayesian hierarchical models, for example, the flexibility of the single-task learning classifier enables the extension to multi-task settings, and the uncertainty on the predicted matrix entries provides us a nature

criterion for acquiring new entries actively.

Our work is in the context of machine learning literature. The finite quadratically-gated mixture of experts (QGME) have been proposed and inferred by expectation-maximization (EM) algorithm in [LLC07a]. Although the model construction of the finite QGME makes the handling of incomplete data easily, it is prone to over-fitting due to ignoring the uncertainty on both model structure and model parameters. Various Dirichlet process mixture models [MEW96, RG02, MO06, SN09, RDG09, HBP10] have been proposed for regression/classification problems with different purposes, but none of them has targeted at the missing data problem. For the matrix-completion, many approaches are based on the singular value decomposition (SVD) [DSG09, LT07, SM07, SM08, LU09], with the challenge of the selection of the number of latent components.

This dissertation makes the following contributions:

- A general robust classifier handling incomplete data is proposed, which is applicable to general classification. As a generalization of the finite QGME via the Dirichlet process, the iQGME proposed here does not require model selection and is unlikely prone to over-fitting. Moreover, its hierarchical structure also makes it appropriate for multi-task settings.
- Incomplete data are handled while learning multiple classification subtasks simultaneously, by further extending the iQGME. As demonstrated on remote sensing and handwriting classification data sets, this extension is important especially for situations with scarce training samples, and with a large portion of features missing.
- A novel Bayesian matrix-completion method is proposed, where the number of decomposed component matrices is inferred explicitly and the model-selection issues are overcome. This model is termed BSVD. Since a binary selecting

variable is introduced for each component matrix, with a zero-favoring prior imposed, our construction is essentially equivalent to minimizing the l_0 norm of the singular values, which saves much computation in a principled way.

- Generalizations of the BSVD model are made to utilize available information more wisely and more efficiently, and to fit the model into a special scenario: data matrices with integer counts as entries. These generalizations could be applied on other matrix-decomposition models without difficulty.

6.2 Future Work

Concerning future research, we note that the use of multi-task learning provides an important class of contextual information, and therefore is particularly useful when one has limited labeled data and when the data are incomplete (missing features). Another form of context that has received significant recent attention is semi-supervised learning [Zhu05]. There has been recent work on integrating multi-task learning with semi-supervised learning [LLC07c]. An important new research direction includes extending semi-supervised multi-task learning to realistic problems for which the data are incomplete.

For the matrix completion, we derive a Bayesian model for general low-rank matrix completion and apply it on collaborative filtering. When we address some specific problem, we may impose more structures accordingly. As we have mentioned, instead of naively selecting most uncertain elements to acquire, for matrices with many entries highly correlated, we should select those elements of \mathbf{Y} that are uncertain *and* uncorrelated. For large-scale data sets, we may dramatically reduce computation expense by only selecting those representative elements and discarding most of noninformative data for model learning. For the MLB data set, which con-

tains records from a large time span, we have ignored any time information and pool all the data together. An interesting direction could be considering time-involving dependence for the BSVD model.

Bibliography

- [ABEV09] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: operator estimation with spectral regularization. *J. Machine Learning Research*, 2009.
- [AC93] J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.
- [Ant74] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152C1174, 1974.
- [Att00] H. Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.
- [AZ05] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- [Bax95] J. Baxter. Learning internal representations. In *Proceedings of the Workshop on Computational Learning Theory (COLT)*, pages 311–320, 1995.
- [Bax00] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.
- [BC01] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *International Conference on Machine Learning (ICML)*, 2001.
- [BC06] T. D. Bie and N. Cristianini. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *Journal of Machine Learning Research*, 7:1409–1436, 2006.

- [Bea03] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD dissertation, University College London, Gatsby Computational Neuroscience Unit, 2003.
- [BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall, 1984.
- [Bis99] C. M. Bishop. Bayesian PCA. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 382–388, 1999.
- [BJ06] D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.
- [BM73] D. Blackwell and J. MacQueen. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1:353–355, 1973.
- [BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *The Annual Conference on Learning Theory (COLT)*, pages 92–100, 1998.
- [BN03] M. Belkin and P. Niyogiy. Semi-supervised learning on manifolds. *Machine Learning Journal, Special Issue on Clustering*, 2003.
- [BP66] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of nite state markov chains. *Ann. Math. Stat.*, 37:1554–1563, 1966.
- [BS94] J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.
- [Car97] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.
- [CHE⁺08] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9:1–21, 2008.
- [CR09] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.
- [CSK01] L. M. Collins, J. L. Schafer, and C. M. Kam. A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, 6(4):330–351, 2001.

- [CST00] M. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge Univ. Press, Cambridge, U.K., 2000.
- [CWS03] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.
- [DeC06] D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proc. of International Conference on Machine Learning (ICML)*, 2006.
- [DHS08] U. Dick, P. Haider, and T. Scheffer. Learning from incomplete data with infinite imputations. In *International Conference on Machine Learning (ICML)*, 2008.
- [DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.
- [DP08] D. B. Dunson and J.-H. Park. Kernel stick-breaking processes. *Biometrika*, 95, 2008.
- [DPP07] D. B. Dunson, N. Pillai, and J.-H. Park. Bayesian density regression. *Journal of the Royal Statistical Society: Series B*, 69, 2007.
- [DSG09] R. Herbrich D. Stern and T. Graepel. Matchbox: Large scale online bayesian recommendations. In *18th International World Wide Web Conference (WWW2009)*, 2009.
- [EBFW07] E. B. Sudderth E. B. Fox and A. S. Willsky. Hierarchical dirichlet processes for tracking maneuvering targets. In *Proceedings of the International Conference on Information Fusion*, 2007.
- [EMP05] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- [EW95] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.
- [Fer73] T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.

- [Fig03] M. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI*, 25(9):1150–1159, 9 2003.
- [FS99] Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 9 1999.
- [GB00] Z. Ghahramani and M. J. Beal. Variational inference for Bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems (NIPS) 12*, pages 449–455. MIT Press, 2000.
- [GH96] Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996.
- [GHRPS90] A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. M. Smith. Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of American Statistical Association*, 85:972–985, 1990.
- [GJ94] Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical report, Massachusetts Institute of Technology, 1994.
- [GKM05] A. E. Gelfand, A. Kottas, and S. N. MacEachern. Bayesian nonparametric spatial modeling with dirichlet process mixing. *Journal of American Statistical Association*, 100:1021–1035, 2005.
- [Gra02] T. Graepel. Kernel matrix completion by semidefinite programming. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 694–699, 2002.
- [GRS] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*, chapter 1 Introducing markov chain monte carlo.
- [GS06] J. E. Griffin and M. F. J. Steel. Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101:179–194, 2006.
- [HBC10] M.D. Hoffman, D.M. Blei, and P.R. Cook. Bayesian nonparametric matrix factorization for recorded music. In *International Conference on Machine Learning*, pages 439–446, 2010.

- [HBP10] L. Hannah, D. Blei, and W. Powell. Dirichlet process mixtures of generalized linear models. In *Artificial Intelligence and Statistics (AISTATS)*, pages 313–320, 2010.
- [HM82] J. Hanley and B. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 1982.
- [Hof04] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22:89115, 2004.
- [Hof09] P.D. Hoff. Simulation of the matrix Bingham-von Mises-Fisher distribution, with applications to multivariate and relational data. *J. Comp. Graph. Statistics*, 2009.
- [Ibr90] J. Ibrahim. Incomplete data in generalized linear models. *Journal of the American Statistical Association*, 85:765–769, 1990.
- [IJ01] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96:161–173, 2001.
- [JGJS99] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Learning in Graphical Models*. 1999.
- [JJ94] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
- [JJNH91] R. A. Jacobs, M. I. Jordon, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
- [Joa99] T. Joachims. Transductive inference for text classification using support vector machines. In *The 16th International Conference on Machine Learning (ICML)*, pages 200–209. Morgan Kaufmann, San Francisco, CA, 1999.
- [Joa03] T. Joachims. Transductive learning via spectral graph partitioning. In *The Twentieth International Conference on Machine Learning (ICML)*, 2003.
- [KCFH05] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Learning sparse bayesian classifiers: multi-class formulation, fast algorithms, and

generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27, 2005.

- [KG07] A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *Proc. AAAI*, 2007.
- [KWT07] K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2796–2801, 2007.
- [KWX⁺05] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo. On semi-supervised classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.
- [Liu07] Q. Liu. *Exploitation of unlabeled data and related tasks in semi-supervised learning*. PhD dissertation, Duke University, Department of Electrical and Computer Engineering, 2007.
- [LJ08] P. Liang and M. I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 584–591, 2008.
- [LLC07a] X. Liao, H. Li, and L. Carin. Quadratically gated mixture of experts for incomplete data classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 553–560, 2007.
- [LLC07b] Q. Liu, X. Liao, and L. Carin. Learning classifiers on a partially labeled data manifold. In *International Conference on Acoustics, Speech and Signal Processing*, 2007.
- [LLC07c] Q. Liu, X. Liao, and L. Carin. Semi-supervised multitask learning. In *Neural Information Processing Systems*, 2007.
- [LP04] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 512–519, 2004.
- [LPJK07] P. Liang, S. Petrov, M. I. Jordan, and D. Klein. The infinite pcfg using hierarchical dirichlet processes. In *In EMNLP 07*, pages 688–697, 2007.
- [LS99] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788791, 1999.

- [LT07] Y. J. Lim and Y. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, 2007.
- [LU09] N.D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.
- [Mac99] S. N. MacEachern. Dependent nonparametric processes. In *Bayesian Statistical Science Section*, pages 50–55. American Statistical Association, 1999.
- [Mac03] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- [Mar04] B. Marlin. Collaborative filtering: A machine learning perspective. Master’s thesis, University of Toronto, 2004.
- [MEW96] P. Müller, A. Erkanli, and M. West. Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, 83:67–79, 1996.
- [MM98] S. N. MacEachern and P. Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7, 1998.
- [MN89] P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.
- [MO06] E. Meeds and S. Osindero. An alternative infinite mixture of Gaussian process experts. In *NIPS 18*, pages 883–890. MIT Press, 2006.
- [MQR04] P. Müller, F. Quintana, and G. Rosner. A method for combining inference across related nonparametric bayesian models. *Journal of the Royal Statistical Society Series B*, 66:735–749, 2004.
- [MS01] M. Meila and J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.
- [NCD07] K. Ni, L. Carin, and D. B. Dunson. Multi-task learning for sequential data via ihmms and the nested dirichlet process. In *Proceedings of the 24th international conference on Machine learning (ICML)*, 2007.
- [Nea93] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993.

- [NHBM98] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 1998.
- [NJ02] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.
- [PR08] O. Papaspiliopoulos and G. O. Roberts. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika Advance Access*, February 6 2008. doi:10.1093/biomet/asm086.
- [QPC07] Y. Qi, J. Paisley, and Lawrence Carin. Dirichlet process HMM mixture models with application to music analysis. In *International Conference on Acoustics, Speech and Signal Processing*, 2007. to appear.
- [RC99] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.
- [RDG08] A. Rodríguez, D. B. Dunson, and A. E. Gelfang. The nested Dirichlet process. *Journal of the American Statistical Association*, 103, 2008.
- [RDG09] A. Rodríguez, D. B. Dunson, and A. E. Gelfang. Bayesian nonparametric functional data analysis through density estimation. *Biometrika*, 96, 2009.
- [RG02] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *NIPS 14*. MIT Press, 2002.
- [RS05] Jason D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *In Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 713–719, 2005.
- [Rub76] D. B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.
- [Rub87] D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, Inc., 1987.
- [SBHM09] V. Sindhwani, S. Bucak, J. Hu, and A. Mojsilovic. A family of non-negativematrix factorizations for one-class collaborative filtering. In *RecSys 09: Recommender based Industrial Applications Workshop*, 2009.

- [SBS06] P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.
- [See01] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2001.
- [Set94] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 1:639–650, 1994.
- [SG02] J. L. Schafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7:147–177, 2002.
- [SJ01] N. Srebro and T. Jaakkola. Sparse matrix factorization for analyzing gene expression patterns. In *Neural Information Processing Systems (NIPS) 2001 Workshop on Machine Learning Techniques for Bioinformatics*, 2001.
- [SJ02] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.
- [SK09] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:1–19, 2009.
- [SKW06] Y. Sun, M. S. Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of International Conference on Data Mining (ICDM)*, 2006.
- [SM07] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, 2007.
- [SM08] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *The 25th International Conference on Machine Learning (ICML)*, 2008.
- [SMC07] A. D. Szlám, M. Maggioni, and R. R. Coifman. Regularization on graphs with function-adapted diffusion processes, 2007. to appear.
- [SN09] B. Shahbaba and R. Neal. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850, 2009.

- [SRJ05] N. Srebro, J.D.M. Rennie, and T.S. Jaakkola. Maximum-margin matrix factorization. In *Proc. Neural Information Processing Systems*, 2005.
- [Sto74] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.
- [STY05] A. Schwaighofer, V. Tresp, and K. Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.
- [SVH05] A. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.
- [Tip00] M. E. Tipping. The relevance vector machine. In T. K. Leen S. A. Solla and K. R. Müller, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 12, pages 652–658. MIT Press, 2000.
- [TMIJB06] Y. W. Teh, M. J. Beal M. I. Jordan, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.
- [TO96] S. Thrun and J. O’Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 489–497, 1996.
- [TS00] N. Tishby and N. Slonim. Data clustering by Markovian relaxation and the information bottleneck method. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.
- [Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [VB02] P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48:165–187, 2002.
- [WACD09] C. Wang, Q. An, L. Carin, and D. Dunson. Multi-task classification with infinite local experts. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1569–1572, 2009.

- [WC05] D. Williams and L. Carin. Analytical kernel matrix completion with incomplete multi-view data. In *Proceedings of the International Conference on Machine Learning (ICML) Workshop on Learning with Multiple Views*, pages 80–86, 2005.
- [WLJF06] X. Wang, A. Li, Z. Jiang, and H. Feng. Missing value estimation for dna microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7:32, 2006.
- [WLX⁺07] D. Williams, X. Liao, Y. Xue, L. Carin, and B. Krishnapuram. On classification with incomplete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):427–436, 2007.
- [WME94] M. West, P. Müller, and M. D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In P. R. Freeman and A. F. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.
- [WR94] S. R. Waterhouse and A. J. Robinson. Classification using hierarchical mixtures of experts. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, pages 177–186, 1994.
- [WR96] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8*, pages 514–520, 1996.
- [XDC07] Y. Xue, D. B. Dunson, and L. Carin. The matrix stick-breaking process for flexible multi-task learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1063–1070, 2007.
- [XJH95] L. Xu, M. I. Jordan, and G. E. Hinton. An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems (NIPS) 7*, pages 633–640, 1995.
- [XLCK07] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.
- [Xue06] Y. Xue. *Multitask Learning with Sensing Applications*. PhD dissertation, Duke University, Department of Electrical and Computer Engineering, 2006.

- [YLZG09] K. Yu, J. Lafferty, S. Zhu, and Y. Gong. Large-scale collaborative prediction using a nonparametric random effects model. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.
- [YST⁺03] K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical Bayes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 616–623, 2003.
- [YST05] K. Yu, A. Schwaighofer, and V. Tresp. Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005.
- [YTY04] K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical Bayesian framework for information filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.
- [ZBL⁺04] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, 2004.
- [ZCY⁺03] Y. Zhang, L. M. Collins, H. Yu, C. Baum, and L. Carin. Sensing of unexploded ordnance with magnetometer and induction data: theory and signal processing. *IEEE Transactions on Geoscience and Remote Sensing*, 41(5):1005–1015, 2003.
- [ZGL03] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *The Twentieth International Conference on Machine Learning (ICML)*, pages 912–919, 2003.
- [ZGY06] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems*, 2006.
- [Zhu05] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Biography

Chunping Wang was born on April 17th, 1979 in Zigong, China. She received her B.S. and M.S. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 2001 and 2003, respectively. In January 2005, she began to study in the Department of Electrical and Computer Engineering at Duke University, where she received a M.S. degree in 2007, and is currently pursuing her Ph.D. degree. Her current research interests include Bayesian statistics and machine learning, with focus on learning with incomplete data, multi-task learning, and collaborative filtering.

Publications

- C. Wang, X. Liao, D. Dunson, and L. Carin, Classification with Incomplete Data Using Dirichlet Process Priors. Submitted to *Journal of Machine Learning Research*.
- M. Zhou, C. Wang, M. Chen, J. Paisley, D. Dunson, and L. Carin. Nonparametric Bayesian Matrix Completion. to appear at *6th Sensor Array and Multichannel Signal Processing Workshop (SAM 2010)*, Israel.
- M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: algorithm and performance bounds, to appear in *IEEE Transactions on Signal Processing*.
- C. Wang, Q. An, L. Carin, and D. Dunson, Multi-task Classification with Infinite Local Experts. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.1569-1572, 2009.
- Q. An, C. Wang, I. Shterev, E. Wang, L. Carin, and D. Dunson, Hierarchical Kernel Stick-Breaking Process for Multi-Task Image Analysis. *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pp. 17-24, 2008.
- D. Williams, C. Wang, X. Liao, and L. Carin, Classification of Unexploded Ordnance Using Incomplete Multisensor Multiresolution Data. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 7, pp. 2364-2373, 2007.
- C. Wang, X. Sun, and J. Jiang, Coupling Effects among Passive Components Used in Power Electronic Device. *Journal of Power Supply*, vol. 1, pp. 422-426, 2003.