University of Technology Sydney

Faculty of Engineering and Information Technology

The Hong Kong Polytechnic University

Department of Computing

Learning Sparse Graphical Models for Data Restoration and Multi-Label Classification

Qiang Li

A thesis submitted in partial fulfilment of the requirements for
the degree of

Doctor of Philosophy

April 2017

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________(Signed)

______Qiang Li______(Name of student)

To my loving parents.

Abstract

Sparse probabilistic graphical models play an important role in structured prediction when the dependency structure is unknown. By inducing sparsity over edge parameters, a typical sparse graphical model can combine structure learning and parameter estimation under a unified optimization framework. In this thesis, we propose three specific sparse graphical models accompanied by their applications in data restoration and multi-label classification respectively.

For the data restoration task, we propose random mixed field (RMF) model to explore mixed-attribute correlations among data. The RMF model is capable of handling mixed-attribute data denoising and imputation simultaneously. Meanwhile, RMF employs a structured mean-field variational approach to decouple continuous-discrete interactions to achieve approximate inference. The effectiveness of this model is evaluated on both synthetic and real-world data.

For the multi-label classification task, we propose correlated logistic model (CorrLog) and conditional graphical lasso (CGL), to learn conditional label correlations. (1) The CorrLog model characterizes pairwise label correlations via scalar parameters, thus effects in an explicit (or direct) fashion. More specifically, CorrLog extends conventional logistic regression by jointly modelling label correlations. In addition, elastic-net regularization is employed to induce sparsity over the scalar parameters that define label correlations. CorrLog can be efficiently learned by regularized maximum pseudo likelihood estimation which enjoys a satisfying generalization bound. Besides, message passing algorithm is applied to solve the multi-label prediction problem. (2) The CGL model further leverages features in modelling pairwise label correlations in terms of parametric functions of the input features, which effects in an implicit (or indirect) fashion. In general, CGL provides a unified Bayesian framework for structure and parameter learning conditioned on input features. We formulate the multi-label prediction as CGL inference problem, which is solved by a mean field variational approach. Meanwhile, CGL learning is efficient after applying the maximum a posterior (MAP) methodology and solved by a proximal gradient procedure. The effectiveness of CorrLog and CGL are evaluated on several benchmark multi-label classification datasets.

Acknowledgements

I feel so grateful to have Prof. Dacheng Tao and Prof. Jane You as my supervisors in UTS and PolyU respectively. I have learned a lot from their immense and extensive knowledge, meticulous attitude to academic research, and genial personalities. Besides, I am grateful to have Dr. Wei Bian and Dr. Richard Yi Da Xu as my de-facto co-supervisors in UTS. I have also learned a lot from their insightful guidance and elaborate suggestions during the second and third years of my PhD candidature. Actually, without those fruitful discussions, I can hardly achieve the research outcomes which constitute the main content of this thesis.

I would like to thank my friends in UTS, including Maoying Qiao, Mingming Gong, Ruxin Wang, Zhibin Hong, Tongliang Liu, Changxing Ding, Meng Fang, Zhe Xu, Shaoli Huang, Hao Xiong, Xiyu Yu, Guoliang Kang, Liu Liu, Chen Gong, Huan Fu, Baosheng Yu, Zhe Chen, Zijing Chen, Yali Du, Xun Yang, Jiankang Deng, Jing Yang, Dr. Jun Li, Dr. Tianyi Zhou, Dr. Yong Luo, Chang Xu, A/Prof. Shengzheng Wang, A/Prof. Xianhua Zeng, Prof. Bo Du, A/Prof. Shigang Liu, A/Prof. Wankou Yang, A/Prof. Xianye Ben, A/Prof. Tao Lei, Liping Xie, Jun Li, Long Lan, Chunyang Liu, Bozhong Liu, Sujuan Hou, Wuxia Yan, Haishuai Wang, Qin Zhang, Zhaofeng Su, Zhiguo Long, Lianyang Ma, Nannan Wang, Fei Gao, Weilong Hou, Mingjin Zhang, Chao Ma and many others. I learned a lot from the discussions and interactions with them in both academic research and daily life. In particular, I really enjoyed the relaxed atmosphere of the afternoon meet-ups with Dr. Wei Bian and Maoying Qiao. Thanks for their understanding and tolerance to my tedious complaints on research, and thanks for their help in my preparation of research papers. Another particular thanks goes to Mingming Gong and Ruxin Wang, thanks for their encouragements and help on my PhD application, comments and suggestions on my research, and considerate assistance in daily life.

I would also like to thank my friends in PolyU, including A/Prof. Lefei Zhang, Ruohan Zhao, Siwei Hu, Yanxin Hu, A/Prof. Risheng Liu, Dr. Xianbiao Qi, Hui Li, Runjie Tan, Wengen Li, Ruosong Yang, Wei Lu, Yumeng Guo, Xiao Shen, Sitong Mao, Jiaxin Chen, Minglei Li, Edison Chan, Qiang Zhang, Liang Zhang, Quanyu Dai, Zimu Zheng, Yu Lei, Lei Han, Lei Xue, Zhijian He, Xingye Lu and many others. Those occasional discussions and talks with them made the fourth year of my PhD study a nice journey.

Finally, my special thanks goes to my family, including my parents, my elder brother, my grandmother and also in memory of my grandfather who left us three years ago. Thanks for their endless love, encouragements, support and blessing that helped me make it through the hard times of my study and life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Probabilistic graphical models (PGMs) are the marriage of graph theory and probability theory, which are very useful to deal with uncertainty and complexity in the design and analysis of machine learning algorithms [12, 58, 133]. In general, PGMs use nodes (empty or filled) to represent random variables, and edges (undirected or directed) to represent dependency between them. Based on the graphical representation, three basic tasks of PGMs are usually carried out: (1) structure learning is to recover the pattern of edges or the structure of the graph that best models or explains the data; (2) parameter learning aims to obtain effective model parameters by maximizing data likelihood; (3) inference targets to efficiently calculate partition function value, marginals and most probable explanation (MPE) solution.

As a recent research interest, sparse PGMs play an important role in structured prediction when the dependency structure is unknown. By inducing sparsity over edge parameters, a typical sparse graphical model can combine structure learning and parameter estimation under a unified optimization framework. According to the types of random variables involved, sparse PGMs can be summarized into three main categories: continuous, discrete and mixed graphical models. Continuous graphical models deal with real-valued random variables which may be modelled, for example, by Gaussian, exponential, gamma, Wishart, beta, Dirichlet and von Mises distributions. Discrete graphical models handle discrete-valued random variables by using discrete distributions such as Bernoulli, binomial, categorical, multinomial and Poisson. Mixed graphical models consider mixed continuous and discrete random variables.

1.2 Continuous Graphical Models

We restrict our discussion on pairwise undirected graphical models (UGMs). In general, given a set of m random variables $\{x_i\}_{i=1}^m$, a pairwise UGM defines the joint distribution $p(\mathbf{x})$ in terms of unary and pairwise potentials,

$$p(\mathbf{x}) = \exp\left\{\sum_{i=1}^{m}\phi(x_i) + \sum_{i<j}\phi_{ij}(x_i, x_j) - A(\Theta)\right\}, \tag{1.1}$$

where $A(\Theta) = \log\exp\left\{\sum_{i=1}^{m}\phi(x_i) + \sum_{i<j}\phi_{ij}(x_i, x_j)\right\}$ is the log-partition function which makes the distribution normalized.

Let the m random variables follow a multivariate Gaussian distribution, the unary and pairwise potentials are in quadratic form of $\mathbf{x}$, the log-partition function can be calculated in closed-form. More specifically,

$$p(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\Omega(\mathbf{x}-\mu) - \log\left((2\pi)^{m/2}(\det\Omega)^{-1/2}\right)\right\}, \tag{1.2}$$

where $\Theta = \{\mu, \Omega\}$ are the model parameters. Given a set of training data $\{\mathbf{x}^{(l)}\}_{l=1}^n$, the structure learning of Gaussian graphical model can be expressed as minimizing the negative log-likelihood with ℓ_1-regularization over the precision matrix entries. The optimization problem is

$$\min_{\Omega \succeq 0} -\log\det\Omega + \mathrm{Tr}(\mathbf{S}\Omega) + \lambda\sum_{i<j}|\omega_{ij}|, \tag{1.3}$$

where $\mathbf{S} = \frac{1}{n}\sum_{l=1}^{n}(\mathbf{x}^{(l)} - \bar{\mathbf{x}})(\mathbf{x}^{(l)} - \bar{\mathbf{x}})^T$ refers to the sample covariance matrix. Note that $\bar{\mathbf{x}} = \frac{1}{n}\sum_{l=1}^{n}\mathbf{x}^{(l)}$ is the sample mean of the training data. The positive semi-definite (PSD) constraint $\Omega \succeq 0$ ensures that the solution yields a valid distribution. In literature, the above optimization problem is usually referred to as the graphical lasso [44, 91, 148]. It was firstly discussed in [91] where the pseudo-likelihood methodology is employed to approximate the original problem and leads to multiple interleaved linear regression sub-problems.

Apart from the Gaussian graphical model, it is also possible to consider other kinds of continuous distributions such as exponential, gamma, beta, Wishart, Dirichlet and von Mises. For example, [106] discussed about the von Mises graphical model in terms of structure learning, parameter estimation and inference which is claimed to be useful in modelling angular data.

1.3 Discrete Graphical Models

By applying ℓ_1-regularization over the parameters of a multivariate discrete distribution, we can obtain the structure learning problem of discrete graphical models [33, 69, 134]. In literature, most of the works restricted the discrete variables to be binary with Ising potentials. Given a set of m binary valued random variables $\{y_i\}_{i=1}^{m}$, the joint distribution of Ising graphical model can be formulated as below

$$p(\mathbf{y}) = \exp\left\{\sum_{i=1}^{m}\beta_i y_i + \sum_{i<j}\alpha_{ij}y_i y_j - A(\Theta)\right\}, \tag{1.4}$$

where $\Theta = \left\{\{\beta_i\}_{i=1}^m, \{\alpha_{ij}\}_{i<j}^m\right\}$ and $A(\Theta)$ is the log-partition function. One can observe that, an edge parameter α_{ij} being zero reflects a missing edge between nodes i and j in the graph. Given a set of training data $\{\mathbf{y}\}_{l=1}^n$, the structure learning of Ising graphical model can be expressed as minimizing the ℓ_1-regularized negative log-likelihood. In other words,

$$\min_{\Theta} - \sum_{i=1}^m \beta_i \bar{\phi}_i - \sum_{i<j} \alpha_{ij} \bar{\phi}_{ij} + A(\Theta) + \lambda \sum_{i<j} |\alpha_{ij}|, \tag{1.5}$$

where $\bar{\phi}_i = \frac{1}{n} \sum_{l=1}^n y_i^{(l)}$ and $\bar{\phi}_{ij} = \frac{1}{n} \sum_{l=1}^n y_i^{(l)} y_j^{(l)}$ are usually referred to as the sufficient statistics of Ising graphical model. Due to the intractable normalizing constant, the above problem is more complicated than that of Gaussian case. To handle the computational intractability, different approximate approaches were investigated in literature. For example, [134] applied the pseudo-likelihood approximation to the original problem which leads to solving multiple interleaved logistic regression sub-problems.

Until now, we mainly focus on the Ising graphical model to build intuition. There are other kinds of discrete graphical models involving categorical, multinomial or Poisson random variables. For example, [143] investigated the Poisson graphical model to discover dependency structure of count data. Meanwhile, it is also possible to consider continuous approximation or repulsion in learning discrete graphical models. For continuous approximation, [82] investigated the relationship between the structure of a discrete graphical model and the support of the inverse of a generalized covariance matrix (which is defined by augmenting the usual covariance matrix with higher-order interaction terms). This work opened a possibility of applying Gaussian graphical model structure learning methods to certain classes of discrete graphical models. For repulsive structure learning, [18] investigated anti-ferromagnetic Ising models and provided a structure learning algorithm whose complexity depends on the strength of repulsion.

1.4 Mixed Graphical Models

By applying group ℓ_1-regularization over the edge parameters among mixed random variables, we can obtain the structure learning problem of mixed graphical models [27, 68, 142]. Consider a set of m_c real valued and m_d binary valued random variables $\{x_s\}_{s=1}^{m_c}, \{y_i\}_{i=1}^{m_d}$, the joint distribution of Gaussian-Ising graphical model [68] can be formulated as

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) = \exp \Bigg\{ &-\frac{1}{2}\mathbf{x}^T \Omega \mathbf{x} + \sum_s \mu_s x_s + \sum_s \sum_i \rho_{si} x_s y_i \\ &+ \sum_i \beta_i y_i + \sum_{i<j} \alpha_{ij} y_i y_j - A(\Theta) \Bigg\}, \end{aligned} \tag{1.6}$$

where $\Theta = \{\Omega, \{\mu_s\}, \{\rho_{si}\}, \{\beta_i\}, \{\alpha_{ij}\}\}$ and $A(\Theta)$ is the log-partition function. Given a set of training data $\{\mathbf{x}, \mathbf{y}\}_{l=1}^n$, the structure learning of Gaussian-Ising graphical model is to minimize the ℓ_1-regularized negative log-likelihood.

$$\begin{aligned} \min_{\Theta:\ \Omega \succeq 0} \ & \frac{1}{2}\mathrm{Tr}(\mathbf{S}\Omega) - \sum_s \mu_s \bar{\phi}_s^c - \sum_s \sum_i \rho_{si} \bar{\phi}_{si}^m - \sum_i \beta_i \bar{\phi}_i^d - \sum_{i<j} \alpha_{ij} \bar{\phi}_{ij}^d \\ & + A(\Theta) + \lambda_c \sum_{s<t} |\omega_{st}| + \lambda_m \sum_s \sum_i |\rho_{si}| + \lambda_d \sum_{i<j} |\alpha_{ij}|, \end{aligned} \tag{1.7}$$

where $\mathbf{S} = \frac{1}{n}\sum_{l=1}^n (\mathbf{x}^{(l)})(\mathbf{x}^{(l)})^T$, $\bar{\phi}_s^c = \frac{1}{n}\sum_{l=1}^n x_s^{(l)}$, $\bar{\phi}_{si}^m = \frac{1}{n}\sum_{l=1}^n x_s^{(l)} y_i^{(l)}$, $\bar{\phi}_i^d = \frac{1}{n}\sum_{l=1}^n y_i^{(l)}$ and $\bar{\phi}_{ij}^d = \frac{1}{n}\sum_{l=1}^n y_i^{(l)} y_j^{(l)}$ are the sufficient statistics. To handle the computational intractability, we can again use the pseudo-likelihood approximation. In literature, mixed graphical models were first proposed by Lauritzen and Wermuth [67] (and further studied in [45, 64–66]). However, the number of model parameters scales exponentially with the number of discrete variables. To reduce the number of model parameters, [68] considered a specialization with only pairwise interactions and fixed precision matrix of continuous variables. As a little more complex specialization, [27] further allowed triple interactions between two discrete and one continuous variable. Motivated by the advantages of exponential families, [142] considered mixed graphical models of general exponential family distributions. Three examples, Gaussian-Ising, Poisson-Ising and Gaussian-Poisson, are discussed to model heterogeneous data.

1.5 Data Restoration

The task of data restoration is to reduce the effect of noise and missing values, which plays a critical preprocessing step in developing complex data mining algorithms. In literature, some researchers describe the data restoration task as a very huge concept, which may include but not limited to outlier detection and removal, noise reduction, and missing value imputation. Interested readers can check more details in the survey paper [158]. It is worth mentioning that outlier detection and removal has been very hot topic for the past decades [110]. The aim is to distinguish and remove those points that are faraway from other points. In fact, outliers can be regarded as instance-level noise, thus the outlier removal process will intrinsically shrink the dataset. In contrast, attribute-level noise refers to undesirable incorrect measurements in some specific attribute of all instances. For example, the Ecoli dataset [79] is used for the prediction of proteins' cellular localization sites and it contains 336 instances with 8 attributes for each instance. The 6-th attribute, which represents the score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins, contains random noise.

Different from the characteristics of noise, missing values are unavailable measurements. In practice, there are various reasons leading to missing values, such as incaution or unwillingness in manual data entry process, equipment errors and too high acquisition cost. Handling missing values highly depends on the task at hand. There are 3 types of missing values: missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR) [80].

MCAR refers to the scenario where missingness of attribute/feature values is independent of both observed and unobserved measurements. MAR occurs when the missingness pattern is conditionally independent of the unobserved features given the observations. Missing data is NMAR when the MAR condition fails to hold.

Complex real-world data often contain continuous and discrete variables measuring different aspects of the same subject. For example, survey reports, social networks and high-throughput genomics. Note that, in machine learning and data mining literature, the word "heterogeneous" may refer to two different research settings: (1) heterogeneity inside a single data source and (2) heterogeneity between multiple data sources. More specifically, the first setting describes a subject consisting of mixed continuous and discrete attributes, thus it is also called "mixed-attribute" by some researchers [100, 157]. As for the second setting, take heterogeneous social networks for example, given a set of subjects, there exist multiple different networks, each representing a particular kind of relationship, and each kind of relationship may play a distinct role in a particular task [21].

In literature, there are several flows of works on dealing with data noise and missing values, such as preprocessing (denoising and imputation), designing reduced models, training robust classifier, etc. We present a brief overview of the research works as below.

1.5.1 Data Denoising Methods

In this subsection, we will try to focus on the less researched attribute-level data denoising methods. Unlike popular image denoising research, attribute-level data denoising research has long been negalected. The possible reason is, an image usually has very strong local smoothness which can benefit denoising, but such local smoothness rarely exists in general data, such as survey report data.

To achieve good local smoothness, [73] proposed to use cluster labels to rearrange all the instances in Ecoli dataset [79]. Then a wavelet shrinkage method is employed to filter certain attribute across instances. Although such “fake” smoothness may occasionally make sense, it is intrinsically unfounded and prone to random instability.

Different from the above deterministic approach, probabilistic models provide a very reasonable platform for handling noisy data. The key assumption is the observed noisy data are generated by adding random noise to those latent noise-free data. Upon this generative perspective, one can define some meaningful prior on latent variables which encodes the attribute-level correlations. In addition, given the latent graphical structure, the observed variables can be thought of independent to each other. In literature, there are many well-studied latent variable models, such as finite mixture models [49], factor analysis and probabilistic principle component analysis (PPCA) [123], mixtures of factor analyzers [48], mixtures of PPCA [122] and their Bayesian extensions [94,95]. Those well-studied models can be straightforwardly applied to handle noisy data. Take PPCA for example, the classical PCA is expressed as the maximum likelihood solution of a probabilistic latent variable model. Noise variance is usually very small compared to signal variance and its energy will reside in those small eigenvalues after PCA projection. By shrinking those small eigenvalues via Wienner filter, one can obtain a “denoised” version of the original data matrix after reconstruction. This is also the key principle behind many state-of-the-art signal/image denoising algorithms [19, 32].

Below we review some of the effective attempts along this approach. [132] addressed the problem of learning a Gaussian mixture from a set of noisy data points. Different from ubiquitous noise case, [61] considered partial noise case. They presented an approach for identifying noisy fields (i.e., some specific attributes) and using the remaining noise-free fields for subsequent modeling and analysis. To achieve such objective, they designed three components for the model: a generative model of the clean records, a generative model of the noise values, and a probabilistic model of the corruption process.

1.5.2 Data Imputation Methods

Data imputation techniques aim at providing good estimates of missing values. From the first sense, missing values can be filled with zero, mean/mode value of available attribute values in database. Other recent techniques modified classical regressors/classifiers to impute continuous/discrete attributes. For example, K-nearest neighbors imputation (KNNI) [124] imputes missing values with the mean/mode of K nearest neighbors for continuous/discrete attributes. Clustering-based imputation [70] first cluster all instances into several cluters and then appy KNNI within each cluster. Local least squares imputation (LLSI) [57] imputes missing values of the target instance as a linear combination of similar instances. Support vector machine imputation (SVMI) [52] is a SVM regression based algorithm to fill in missing data. Multiple kernel learning imputation (MKLI) [157] can impute mixed-attribute missing values by estimating kernel density from available attributes in all instances. [117] presented a random forest modification for mixed-attribute data imputation.

Another family of imputation methods builds probabilistic latent variable models to find the most probable completion. More specifically, [49] addressed the imputation problem by learning mixture models from a set of incomplete data. To well handle missing values, the authors derived a suitable expectation-maximization (EM) algorithm. [113] designed a regularized expectation maximization imputation (REMI) method by modeling the latent variable as a multivariate Gaussian. Singular value decomposition imputation (SVDI) [124] combined principle component (PC) regression and EM estimation to estimate missing values. In addition, some researchers developed Bayesian extension to classical latent variable models. For example, [98] utilized Bayesian principle component analysis (BPCA) to conduct PC regression, baysian estimation and EM learning.

1.5.3 Model Induction with Corrupted Data

The previous subsections summarized preprocessing treatments for noise and missing values, this subsection reviews related works on model induction with corrupted data. For model induction with corrupted data, it is usually assumed that the corruption arises similarly both in training and testing data. The objective is to directly train robust classifiers to classify corrupted data, thus the preprocessing step is avoided.

Model induction from noisy data has been widely considered in various topics, [103] studied the effect of noise on the discovery of classification rules and on their accuracy. A modified form of an existing rule-building algorithm that can tolerate noisy descriptions is also presented. [112] investigated the induction of a concept description given noisy instances and under potential concept drift. They presented a solution to the task and claimed it is supported by psychological and mathematical results. The work of [1] recommended the strategy of selecting the most consistent rule for the sample when errors emerge randomly. They also presented an analysis on the estimation of the rate of noise. Based on the statistical reasoning, [9] proposed a novel formulation of support vector classification (TSVC), which allows uncertainty in input data. A probabilistic support vector machine (PSVM) is proposed in [71], to capture the probabilistic information of the separating margin and formulate the decision function within such a noisy environment. [54] replaced the hinge loss of standard SVM by pinball loss, which achieved noise-insensitivity and re-sampling stability.

Model induction from incomplete data also has been widely investigated by extending standard models, such as linear discriminant analysis (LDA), logistic regression, neural networks, support vector machines (SVM) and kernel methods [89]. Following are just some representative works on extending standard SVM formulation to handle missing values. In [8], SVM with certainty is extended via replacing the linear constraints by a probabilistic one. In addition, the model parameters are estimated by means of EM algorithm. [115] presented treatments for dealing with missing variables in the context of Gaussian processes (GP) and SVM. They casted kernel methods as an estimation problem in exponential families, where estimation with missing variables is formulated as computing marginal distributions. [101] defined a modified risk to analyse SVM involving missing values. [24,25] considered the structurally incomplete data case where certain feature values are undefined for some data cases. Unlike those "imputation + classifier" approaches, an instance-specific max-margin framework is formulated to handle instances with missing values. Two approaches, one approximation to standard quadratic program (QP) and another iterative projection method, are designed to solve the resulting difficult optimization problem. [109] designed both online and batch algorithms which are robust to missing values. For batch setting, they developed a convex relaxation of a non-convex problem to jointly estimate an imputation function, used to fill in the values of missing features, along with the classification hypothesis.

The objective of learning against adversary is to train robust models against various simulated corruptions (noise and missing values) and then apply the model to corrupted testing data. As discussed in [34,86], the learning algorithms should anticipate the actions of the adversary and account for them when training a classifier. Notice that, the experimental setting is very similar to our setting, i.e., training data is noise-free and complete, but testing data maybe corrupted with noise and missing values. However, there is an important difference: most of the current data restoration methods (including our algorithm) are unsupervised, while learning against adversary methods are intrinsically belong to supervised learning.

1.6 Multi-Label Classification

The task of multi-label classification is to assign multiple possible labels to a single instance, which is very common in document summarization, music annotation, image classification, and bioinformatics applications. Since this topic maintains a huge literature, we cannot cover all the methods but try to build intuitions using representative methods from two different views.

1.6.1 The View of Label Dependence

As a first attempt, multi-label classification can be naively decomposed into multiple independent single-label classifications which is called the binary relevance (BR) method. Though simple and efficient, BR may perform poorly if labels correlates to each other. To improve the classification performance, it is beneficial to consider correlations among different class labels. For example, in image classification, the existence of a table in an image will probably indicate the existence of a chair.

According to [35, 36], there are two types of label correlations, unconditional and conditional correlations respectively. Roughly speaking, the former characterizes the global label correlations independent of any specific instance, while the latter describes the label correlations conditioned on a specific instance in a local way. Quite a number of multi-label classification algorithms have been proposed in the past a few years, by exploiting either of unconditional or conditional label correlations. From a classification perspective, proper utilization of unconditional correlations can be beneficial but in an average sense due to the marginalization effect. In contrast, modelling of conditional correlations is preferable since they are directly related to prediction. We discuss this view of summarization in Section 3.2 of Chapter 3.

1.6.2 The View of Learning Strategy

Similar to the taxonomy of [50, 125, 153], multi-label classification methods can also be summarized into four main categories according to the view of learning strategy, i.e., problem transformation, algorithm adaptation, dimension reduction, and structure learning.

Problem transformation methods reformulate multi-label classification into single-label classification by virtue of fitting data to algorithm. Methods in this category rely on the basic assumption that single-label classifiers work more effectively in the transformed space. Algorithm adaptation methods extend typical classifiers to multi-label situation by the philosophy of fitting algorithm to data. This category requires specific strategies when modifying typical classifiers, thus an effective strategy for one classifier may not generalize well to another classifier. Dimension reduction methods target to handle high-dimensional features and labels. Structure learning methods leverage label dependency structure estimation to improve multi-label classification. We discuss this view of summarization in Section 4.2 of Chapter 4.

1.6.3 Music Annotation and Retrieval

As a direct application of multi-label classification, music annotation and retrieval deals the specific problem of annotating and retrieving relevant descriptions of a music file. More specifically, music prediction tasks include tags prediction given a song file (a clip or a whole song), artist name prediction, relevant song prediction given a song file, tags or artist name. To exploit semantic relationship between these different musical concepts, several ingredients are required including music representation and semantic correlation modelling. Below we just introduce some of the representative methods, interested readers are referred to a recent survey on music annotation [46].

For music representation, the “bag-of-words” model [51] is typically utilized to build dictionary-based representation of the delta Mel-Frequency Cepstral Coefficient (MFCC) feature. In particular, k-means is firstly applied to MFCC features to learn K cluster centroids which are literally referred to as "audio dictionary". Then for each music file, by counting the number of its MFCC features according to the dictionary, we can obtain its "bag-of-words" (also called vector quantized) representation. Along the line of research, [128] investigated training Gaussian mixture models (GMM) over an audio feature space for each word in a vocabulary. Base on the vector quantized representation, [51] proposed a Codeword Bernoulli Average (CBA) model that learns to predict the probability that a word applies to a song from audio. It is also possible to consider Dirichlet mixtures (DirMix) [92] to model the audio feature space. In [31], the authors proposed the hierarchical EM Gaussian mixture models (HEM-GMM) and the hierarchical EM dynamic texture model (HEM-DTM) to handle multi-modal and complex dependence among data. For semantic correlation modelling, context-SVM [97] investigated how stacked generalization can be utilized to improve the performance of a basic automatic music tag annotation system based on audio content analysis.

1.6.4 Multi-Label Image Classification

Under the generic scope of multi-label classification, multi-label image classification handles the specific problem of predicting the presence or absence of multiple object categories in an image. Like many related high-level vision tasks such as object detection [131, 145], object recognition [5, 136], visual tracking [90, 138], image annotation [43] and scene classification [15, 116], multi-label image classification [83–85, 118, 139, 149] is very challenging due to large intra-class variation. In general, the variation is caused by viewpoint, scale, occlusion, illumination, semantic context, etc.

In literature, many effective image representation schemes have been developed to handle this high-level vision task. Most of the classical approaches derive from handcrafted image features, such as GIST [99], dense SIFT [13], VLAD [55], and object bank [72]. In contrast, the very recent deep learning techniques have also been developed for image feature learning, such as deep CNN features [23,60]. These techniques are more powerful than classical methods when learning from a very large amount of labeled and unlabeled images.

Apart from powerful image representation methods, label correlations have also been exploited to significantly improve image classification performance. Most of the current multi-label image classification algorithms are motivated by considering label correlations conditioned on image features, thus intrinsically falls into the CRFs framework. For example, probabilistic label enhancement model (PLEM) [77] designed to exploit image label co-occurrence pairs based on a maximum spanning tree construction and a piecewise procedure is utilized to train the pairwise CRFs [63] model. More recently, clique generating machine (CGM) [121] proposed to learn the image label graph structure and parameters by iteratively activating a set of cliques. It also belongs to the CRFs framework, but the labels are not constrained to be all connected which may result in isolated cliques.

1.7 Summary of Contributions

In this thesis, we propose three specific sparse graphical models to discover mixed-attribute correlations and conditional label dependency for data restoration and multi-label classification respectively.

In Chapter 2, a random mixed field (RMF) model is proposed to explore mixed-attribute correlations among data. The RMF model employs Gaussian-Potts potential to fit mixed numerical and categorical data. The learned generic RMF prior is capable of handling mixed-attribute data denoising and imputation in a single framework. The content of this chapter is based on our recent publication in [74].

In Chapter 3, an elastic-net correlated logistic (CorrLog) model is developed to learn conditional label correlations in an explicit (or direct) way. The CorrLog model jointly learns multiple logistic regressions and their label correlations measured by scalar parameters. In addition, elastic-net regularization is utilized to balance stability and sparsity over the scalar parameters when exploiting label correlations. This chapter originates from our recent publication in [76].

In Chapter 4, a conditional graphical lasso (CGL) model is proposed to learn conditional label correlations in an implicit (or indirect) fashion. The CGL model leverages features in modelling pairwise label correlations by using parametric functions of the input features. Technically speaking, CGL provides a unified framework for structure learning, parameter estimation and inference from a probabilistic perspective. The multi-label prediction is formulated as CGL inference problem that is solved based on mean field assumption. Meanwhile, label correlation discovery is achieved by CGL learning which is an efficient proximal gradient procedure with the maximum a posterior (MAP) methodology. This chapter is based on our recent publication in [75].

Chapter 2

Random Mixed Field Model for Data Restoration

Noisy and incomplete data restoration is a critical preprocessing step in developing effective learning algorithms, which targets to reduce the effect of noise and missing values in data. By utilizing attribute correlations and/or instance similarities, various techniques have been developed for data denoising and imputation tasks. However, current existing data restoration methods are either specifically designed for a particular task, or incapable of dealing with mixed-attribute data. In this chapter, we develop a new probabilistic model to provide a general and principled method for restoring mixed-attribute data. The main contributions of this study are two-fold: a) a unified generative model, utilizing a generic random mixed field (RMF) prior, is designed to exploit mixed-attribute correlations; and b) a structured mean-field variational approach is proposed to solve the challenging inference problem of simultaneous denoising and imputation. We evaluate our method by classification experiments on both synthetic data and real benchmark datasets. Experiments demonstrate, our approach can effectively improve the classification accuracy of noisy and incomplete data by comparing with other data restoration methods.

2.1 Introduction

Real world data usually contain noise and missing values, which could severely degrade the performance of learning algorithms [88].The task of data restoration is to reduce the effect of noise and missing values, and plays a critical preprocessing step in developing effective learning algorithms. Attribute-level noise and missing values are two of the major concerns in data restoration. In the literature, attribute-level noise refers to undesirable incorrect measurements in some specific attribute of all instances. Different from the characteristics of noise, missing values are unavailable measurements. In practice, there are various reasons leading to noise and missing values, such as incaution or unwillingness in manual data entry process, equipment failure and high acquisition cost.

Data denoising targets to estimate the true value from noisy measurements based on certain assumptions. Unlike popular image denoising research [19, 20], the research on attribute-level data denoising has long been limited. One possible reason is that images usually have strong local smoothness which can benefit denoising, but such local smoothness rarely exists in general data, such as survey reports. To achieve good local smoothness, [73] proposed to use cluster labels to rearrange all the instances in Ecoli dataset [79]. Then a wavelet shrinkage method is employed to filter certain attribute across instances. Although such "ad hoc" smoothness could make sense, it is intrinsically unfounded and prone to random instability. Different from deterministic approaches, probabilistic models provide a more rational approach for handling noisy data. The key assumption is the observed corrupted data are generated by adding random noise to latent noise-free data. Through such generative models, one can exploit informative priors over latent variables so as to encode attribute correlations.

Data imputation aims at providing good estimates of missing values. Deterministic approaches resort to modify classical regressors/classifiers to impute missing attributes. For example, K nearest neighbors imputation (KNNI) [124] imputes missing values with the mean/mode of K nearest neighbors for continuous/discrete attributes, as well as other techniques including local least squares imputation (LLSI) [57], support vector machine imputation (SVMI) [52], multiple kernel learning imputation (MKLI) [157], and random forest imputation [117]. On the other hand, probabilistic latent variable models are employed to find the most probable imputation. For example, [49] addressed the imputation problem by learning mixture models from an incomplete dataset. [113] designed a regularized expectation-maximization imputation (REMI) method by modelling the latent variables with multivariate Gaussian. Singular value decomposition imputation (SVDI) [124] combined principle component (PC) regression and EM estimation to estimate missing values. Some researchers also developed Bayesian principle component analysis based imputation [98], which jointly conducts PC regression, Bayesian estimation and EM learning.

Despite their effectiveness in exploiting attribute correlations and/or instance similarities, existing methods have two main limitations: (1) they are specifically designed for a particular task, either denoising or imputation; (2) most of them are incapable of dealing with mixed-attribute data directly, and a prerequisite conversion step can inevitably cause information loss. In this study, we formulate the mixed-attribute data restoration problem with a random mixed field (RMF) model. Moreover, to solve the resulting challenging inference problem, we derive a structured variational approach based on the mean field assumption. By exploiting mixed-attribute correlations, the proposed framework is capable of mixed-attribute data denoising and imputation at the same time.

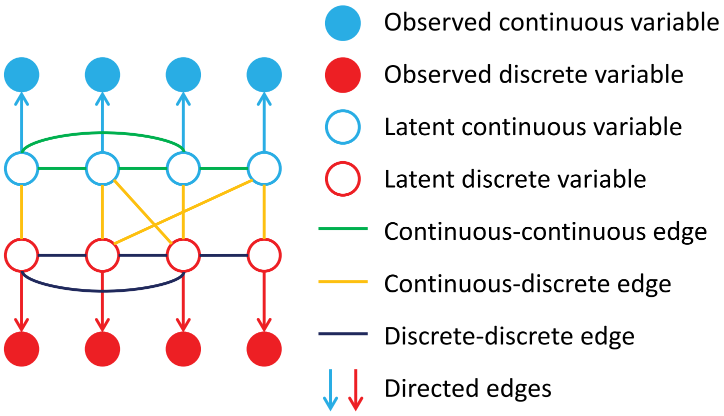
The rest of this chapter is organized as follows. Section 2.2 briefly reviews related research status of mixed graphical models. Section 2.3 introduces the proposed RMF model and illustrates its properties with interpretations. Section 2.4 presents algorithms for RMF inference by a structured mean-field variational approach, and for RMF learning by maximum pseudo likelihood estimation with sparse regularization. Section 2.5 reports results of empirical evaluations, where both synthetic dataset and benchmark real datasets are used, and applications to mixed-attribute data restoration and classification are also considered.

2.2 Related Works

Recently, mixed graphical models have attracted increasing attentions [27,68,142] to meet the need for heterogeneous multivariate data modelling and analysis [37,78,135]. In general, mixed graphical models extend classical graphical models by letting nodes to emerge from different kinds of both continuous and discrete random variables.

In the literature, mixed graphical models were first proposed in [67] to model mixed continuous and discrete variables. In this seminal work, the multinomial and conditional Gaussian distributions are used to represent the joint heterogeneous multivariate distribution. However, the number of model parameters scales exponentially with the number of discrete variables. To reduce the number of model parameters, [68] considered only pairwise interactions and fixed precision matrix for continuous variables. [27] further explored triple interactions between two discrete and one continuous variable. [142] considered mixed graphical models via a unified exponential family distribution to handle mixed-attribute data.

Though an RMF model belongs to general mixed graphical models, we propose to investigate the inference and parameter learning aspects of RMF model which is indeed complementary to the latest structure learning research. Specifically, 1) a structured mean field approach is derived to solve the inference problem of RMF model; 2) a variational expectation maximization algorithm is implemented to estimate the noise parameters given a fixed RMF prior.


Observed continuous variable
Observed discrete variable
Latent continuous variable
Latent discrete variable
Continuous-continuous edge
Continuous-discrete edge
Discrete-discrete edge
Directed edges

(a) Mixed-net (b) Symbol explanations

Figure 2.1: An example of random mixed field model. (a) The hidden network is a “mixed-net” consisting of both continuous and discrete nodes. (b) explains all the four types of nodes and five types of edges.

2.3 Random Mixed Field Model

An RMF model is usually constructed by a hidden network playing the prior part and a corresponding set of observed nodes playing the likelihood part. See Figure 2.1 for a general example of RMF model. Note that, RMF model can be regarded as a specification of general mixed graphical models. In the following, we first describe the general framework of RMF model, and then give derivations of the inference algorithm. Parameter learning and data restoration algorithms will also be discussed. To simplify discussion, we will consider a fully-connected, pairwise, and continuous-discrete mixed graph in the next part of the paper.

Given a general mixed pairwise graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we have the vertex set $\mathcal{V} = \mathcal{V}_u \cup \mathcal{V}_v \cup \mathcal{V}_x \cup \mathcal{V}_y$ representing latent continuous/discrete, observed continuous/discrete variables, and the edge set $\mathcal{E} = \mathcal{E}_{uu} \cup \mathcal{E}_{vv} \cup \mathcal{E}_{uv} \cup \mathcal{E}_{ux} \cup \mathcal{E}_{vy}$ denoting the union of continuous-continuous, discrete-discrete, continuous-discrete connections and emissions. Consider the mixed-net example in Figure 2.1a which consists of four types of nodes and five types of edges. In detail, $\mathcal{V}_u$ and $\mathcal{V}_v$ are represented by cyan and red circles, $\mathcal{V}_x$ and $\mathcal{V}_y$ are denoted by cyan and red filled circles. On the other hand, $\mathcal{E}_{uu}$, $\mathcal{E}_{vv}$ and $\mathcal{E}_{uv}$ correspond to green, purple and yellow line segments; $\mathcal{E}_{ux}$ and $\mathcal{E}_{vy}$ correspond to cyan and red directed line segments.

An RMF model defines a joint distribution over the latent and observed variables according to some specific graphical configuration. In general, the joint distribution can be factorized into the prior and likelihood parts as below,

$$p(u, v, x, y|\Theta) = p(u, v|\Theta_p)p(x, y|u, v; \Theta_n), \tag{2.1}$$

where $\Theta = \Theta_p \cup \Theta_n$ represents the union of prior and noise parameters.

The prior distribution is defined over latent variables via a Gaussian-Potts mixed potential,

$$\begin{aligned} p(u, v|\Theta_p) \propto \exp \Bigg(& \sum_{s=1}^{m_c} \sum_{t=1}^{m_c} -\frac{1}{2}\omega_{st}u_s u_t + \sum_{s=1}^{m_c} \mu_s u_s \\ & + \sum_{s=1}^{m_c} \sum_{j=1}^{m_d} \rho_{sj}(v_j)u_s + \sum_{j=1}^{m_d} \sum_{k=1}^{m_d} \phi_{jk}(v_j, v_k) \Bigg), \end{aligned} \tag{2.2}$$

where $\Theta_p = \{\{\omega_{st}\}, \{\mu_s\}, \{\rho_{sj}\}, \{\phi_{jk}\}\}$ denotes the prior parameters. In particular, ω_{st}, μ_s, ρ_{sj} and ϕ_{jk} parameterizes continuous-continuous edge potential, continuous node potential, continuous-discrete edge potential, and discrete-discrete edge potential, respectively. Upon this mixed Gaussian-Potts prior distribution, we can also obtain node-wise conditional distributions for each variable. Specifically, the conditional distribution of a continuous variable u_s given all its neighboring variables is a Gaussian distribution with a linear regression model for the mean and ω_{ss}^{-1} being the unknown variance,

$$p(u_s|u_{\backslash s}, v) = \frac{\sqrt{\omega_{ss}}}{\sqrt{2\pi}} \exp(\zeta), \tag{2.3}$$
$$\zeta = \frac{-\omega_{ss}}{2} \left(u_s - \frac{\left(\mu_s + \sum_j \rho_{sj}(v_j) - \sum_{t \neq s} \omega_{st} u_t\right)}{\omega_{ss}} \right)^2 .$$

Note that, the backslash operator $\backslash$ is used to exclude variable s in defining the set of neighboring variables. The conditional distribution of a discrete variable v_j given its neighbors is a multinomial distribution with L_j states,

$$p(v_j|v_{\backslash j}, u) = \frac{\exp(\xi_{v_j})}{\sum_{l=1}^{L_j} \exp(\xi_l)}, \tag{2.4}$$
$$\xi_l = \left(\sum_s \rho_{sj}(l) u_s + \phi_{jj}(l, l) + \sum_{k \neq j} \phi_{jk}(l, v_k) \right) .$$

The likelihood is defined based on the assumption that all observed variables are independent to each other conditioned on the latent variables,

$$p(x, y|u, v; \Theta_n) = \prod_{s=1}^{m_c} p(x_s|u_s) \prod_{j=1}^{m_d} p(y_j|v_j), \tag{2.5}$$

where $\Theta_n = \{\{\sigma_s\}, \{\varphi_j\}\}$ denotes the noise parameters of Gaussian and multinomial distributions. In other words, the continuous emission corresponds to additive white Gaussian noise (AWGN), and the discrete emission represents random flipping noise (RFN). Consequently, the distribution of x_s conditioned on u_s is modelled as a Gaussian with the noise parameter σ_s,

$$p(x_s|u_s) = \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left(-\frac{1}{2\sigma_s^2} (x_s - u_s)^2 \right) . \tag{2.6}$$

And the distribution of y_j given v_j is modelled as a multinomial distribution

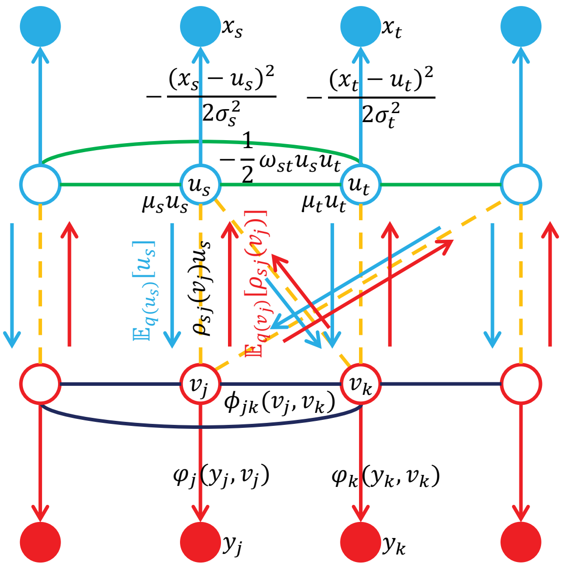

x_s
x_t
$-\frac{(x_s-u_s)^2}{2\sigma_s^2}$
$-\frac{(x_t-u_t)^2}{2\sigma_t^2}$
$-\frac{1}{2}\omega_{st}u_su_t$
u_s
u_t
μ_su_s
μ_tu_t
$\mathbb{E}_{q(u_s)}[u_s]$
$\rho_{sj}(v_j)u_s$
$\mathbb{E}_{q(v_j)}[\rho_{sj}(v_j)]$
v_j
v_k
$\phi_{jk}(v_j,v_k)$
$\varphi_j(y_j,v_j)$
$\varphi_k(y_k,v_k)$
y_j
y_k

Figure 2.2: The proposed structured mean field approximation can be regarded as cutting off those mixed-type edges and absorbing the interactions in the form of expected sufficient statistics, i.e., $\mathbb{E}_{q(u_s)}[u_s]$ and $\mathbb{E}_{q(v_j)}[\rho_{sj}(v_j)]$, respectively. Such a posterior approximation will result in two separate subgraphs, which are much easier to handle. In addition, it is required to alternately update each of the two subgraphs' joint distributions until convergence.

parameterized by noise parameter φ_j,

$$p(y_j|v_j) = \frac{\exp(\varphi_j(y_j, v_j))}{\sum_{l=1}^{L_j} \exp(\varphi_j(l, v_j))}. \tag{2.7}$$

2.4 Algorithms

2.4.1 Structured Mean Field

With an RMF model, data restoration can be achieved by the inference over posterior distribution $p(u, v|x, y; \Theta)$. Since the calculation of the likelihood $p(x, y; \Theta)$ is intractable, we seek to approximate inference approaches. Specifically, we use the variational approach, which is considered to be more efficient than sampling methods. Based on the mean field assumption, the optimal variational approximation of $p(u, v|x, y; \Theta)$ is given by

$$q^*(u, v) = \arg \min_{\substack{q(u,v)= \\ q(u)q(v)}} \mathrm{KL}[q(u, v) \| p(u, v|x, y; \Theta)]. \tag{2.8}$$

The minimization of the Kullback-Leibler divergence in (2.8) can be achieved by maximizing a lower bound,

$$\mathcal{L}(q) = \mathbb{E}_{q(u)q(v)} \left[\ln \frac{p(x, y, u, v)}{q(u)q(v)} \right] \tag{2.9}$$

of the log evidence $\ln p(x, y) = \ln \sum_v \int_u p(x, y, u, v)$ w.r.t. $q(u)$ and $q(v)$, respectively. Accordingly, the update formula for $q(u)$ and $q(v)$ are given by [12],

$$q(u) \leftarrow \frac{1}{Z_u} \exp \mathbb{E}_{q(v)}[\ln p(u, v, x, y)] \tag{2.10}$$

$$q(v) \leftarrow \frac{1}{Z_v} \exp \mathbb{E}_{q(u)}[\ln p(u, v, x, y)], \tag{2.11}$$

where $\mathbb{E}_p[f]$ calculates the expectation of function f w.r.t. distribution p, and Z_u and Z_v are the normalization terms.

To solve Eqn. (2.10) for updating $q(u)$, we evaluate the expectation w.r.t. $q(v)$,

$$\begin{aligned}
&\mathbb{E}_{q(v)}[\ln p(u, v, x, y)] \equiv -\frac{1}{2} \sum_{s=1}^{m_c} \sum_{t=1}^{m_c} \omega_{st} u_s u_t \\
&+ \sum_{s=1}^{m_c} \mu_s u_s + \sum_{s=1}^{m_c} \sum_{j=1}^{m_d} \mathbb{E}_{q(v_j)}[\rho_{sj}(v_j)] u_s \\
&+ \sum_{s=1}^{m_c} \left[-\frac{(x_s - u_s)^2}{2\sigma_s^2} \right] + \sum_{j=1}^{m_d} \mathbb{E}_{q(v_j)}[\varphi_j(y_j, v_j)] \\
&\equiv -\frac{1}{2} u^T \hat{\Omega} u + \hat{\gamma}(v, x)^T u, \qquad (2.12)
\end{aligned}$$

where the notation $\equiv$ denotes the two terms on the left and right hand sides are equivalent up to a constant, and $\Omega = \{\omega_{st}\}$, $\hat{\Omega} = \Omega + \mathrm{diag}\{\frac{1}{\sigma_s^2}\}$, $\{\hat{\gamma}(v, x)\}_s =$

$\mu_s + \sum_j \mathbb{E}_{q(v_j)}[\rho_{sj}(v_j)] + \frac{x_s}{\sigma_s^2}$. Fortunately, $q(u)$ follows a multivariate Gaussian distribution, $q(u) = \mathcal{N}(u|\hat{B}^{-1}\hat{\gamma}(v,x), \hat{B}^{-1})$. Notice that, for the Gaussian-Potts model defined in Eqn. (2.2), we do not need to calculate the inverse of the updated precision matrix $\hat{B}$. The reason is that the continuous-discrete edge potentials are only absorbed into the first-order term of $q(u)$. In addition, the noise term $\mathrm{diag}\{\frac{1}{\sigma_s^2}\}$ is added to the diagonal of B thus does not affect the original graphical connections defined in B. Consequently, algorithms such as Gauss elimination and GaBP [11] can be employed to efficiently infer the mean $\hat{B}^{-1}\hat{\gamma}(v,x)$ when B is sparse.

Regarding Eqn. (2.11), we have the expectation w.r.t. $q(u)$,

$$\begin{aligned}
&\mathbb{E}_{q(u)}[\ln p(u,v,x,y)] \\
&\equiv \sum_{j=1}^{m_c}\sum_{k=1}^{m_c} \phi_{jk}(v_j,v_k) + \sum_{j=1}^{m_d}\sum_{s=1}^{m_c} \rho_{sj}(v_j)\mathbb{E}_{q(u_s)}[u_s] \\
&+ \sum_{s=1}^{m_c} \mathbb{E}_{q(u_s)}\left[-\frac{(x_s-u_s)^2}{2\sigma_s^2}\right] + \sum_{j=1}^{m_d} \varphi_j(y_j,v_j) \\
&\equiv \sum_{j=1}^{m_d}\sum_{k=1}^{m_d} \phi_{jk}(v_j,v_k) + \sum_{j=1}^{m_d} \hat{\varphi}_j(y_j,v_j,u), \qquad (2.13)
\end{aligned}$$

where $\hat{\varphi}_j(y_j,v_j,u) = \sum_s \rho_{sj}(v_j)\mathbb{E}_{q(u_s)}[u_s] + \varphi_j(y_j,v_j)$. In other words, $q(v)$ follows a pairwise discrete MRF, $q(v) \propto \exp\left\{\sum_j \sum_k \phi_{jk}(v_j,v_k) + \sum_j \hat{\varphi}_j(y_j,v_j,u)\right\}$. Note that, for the Gaussian-Potts model defined in Eqn. (2.2), those interaction and emission terms $\{\hat{\varphi}_j\}$ do not affect the original graphical connection defined in $\{\phi_{jk}\}$. Thus, we can use the loopy belief propagation [93] algorithm to solve the pairwise discrete MRF inference problem.

In the above derivations, the mixed-type edge terms appeared in both $q(u)$ and $q(v)$ but in different forms. Figure 2.2 illustrates the process of formulating two interacting subgraphs via $\mathbb{E}_{q(u_s)}[u_s]$ and $\mathbb{E}_{q(v_j)}[\rho_{sj}(v_j)]$ when the hidden network is a mixed-net. The alternative updating between $q(u)$ and $q(v)$ is performed until convergence to a stationary point. The stationary point corresponds to two completely independent subgraphs that jointly approximate the whole mixed graph.

2.4.2 Parameter Estimation

Considering the data restoration task, we follow the setting of a clean training dataset and a corrupted testing dataset [111]. Regarding "clean", we mean the samples are noise-free and complete and "corrupted" means the samples contain noise and missing values. According to this setting, the RMF prior parameters Θ_p can be learned from the clean training dataset. Fortunately, several third-party learning techniques, such as, variants of graphical lasso [44], ℓ_1 regularized pseudo-likelihood [6, 68] and ℓ_1 regularized node-wise regression [141], can be utilized to learn this generic prior. Although these techniques are originally designed for structure learning, the resulting sparsified parameters will not only indicate the graphical structure, but also provide a good parameterization of the generic prior.

When restoring the corrupted testing dataset, domain knowledge can be employed to yield a good estimate of the noise parameters Θ_n. If unfortunately this method fails, a variational EM algorithm can be adopted to estimate noise parameters given all testing data and the generic RMF prior. In general, given N i.i.d. observation samples $X = \{x^{(i)}\}$ and $Y = \{y^{(i)}\}$, the variational EM algorithm iterates between variational inference (E-step) and parameter estimation (M-step). Since the RMF prior parameters are fixed after learning from the training dataset, we can only iteratively infer $q(u, v)$ and estimate Θ_n on the testing dataset until convergence. The corresponding noise parameter estimation in M-step is achieved by taking derivatives of the objective function $\mathcal{Q}(\Theta)$

$$\begin{aligned}\mathcal{Q}(\Theta) &= \sum_i \sum_{v^{(i)}} \int_{u^{(i)}} q(u^{(i)}, v^{(i)}) \ln p(x^{(i)}, y^{(i)}, u^{(i)}, v^{(i)}) \\ &= \sum_i \sum_{v^{(i)}} \int_{u^{(i)}} q(u^{(i)}) q(v^{(i)}) \Big[\ln p(u^{(i)}, v^{(i)}) \\ &+ \sum_s \ln p(x_s^{(i)} | u_s^{(i)}) + \sum_j \ln p(y_j^{(i)} | v_j^{(i)}) \Big] \end{aligned} \tag{2.14}$$

w.r.t. σ_s^2 and $\varphi_j(a, b)$ respectively. Note that we have made explicit the testing sample index i for clarity.

For continuous noise parameter, since σ_s^2 only appears in $p(x_s^{(i)} | u_s^{(i)})$, we can throw all other terms into the constant. After applying the continuous likelihood distribution in Eqn. (2.6), the specified objective of Eqn. (2.14) is

$$\begin{aligned}\mathcal{Q}(\sigma_s^2) &\equiv \sum_i \sum_{v^{(i)}} \int_{u^{(i)}} q(u^{(i)}) q(v^{(i)}) \ln p(x_s^{(i)} | u_s^{(i)}) \\ &= \sum_i \int_{u_s^{(i)}} q(u_s^{(i)}) \ln p(x_s^{(i)} | u_s^{(i)}) \\ &\equiv -\frac{N}{2} \ln \sigma_s^2 - \frac{1}{2\sigma_s^2} \sum_i \int_{u_s^{(i)}} q(u_s^{(i)}) (x_s^{(i)} - u_s^{(i)})^2 \end{aligned} \tag{2.15}$$

Taking derivative of $\mathcal{Q}(\sigma_s^2)$ w.r.t. σ_s^2, and setting it to zero, we have

$$\begin{aligned} -\frac{N}{2\sigma_s^2} + \frac{1}{2\sigma_s^4} \sum_i \int_{u_s^{(i)}} q(u_s^{(i)}) (x_s^{(i)} - u_s^{(i)})^2 &= 0 \\ \Rightarrow \quad \sigma_s^2 &= \frac{1}{N} \sum_i \int_{u_s^{(i)}} q(u_s^{(i)}) (x_s^{(i)} - u_s^{(i)})^2 \\ &= \frac{1}{N} \sum_i (x_s^{(i)})^2 - \frac{2}{N} \sum_i x_s^{(i)} \mathbb{E}_{q(u_s^{(i)})}[u_s^{(i)}] \\ &\quad + \frac{1}{N} \sum_i \mathbb{E}_{q(u_s^{(i)})}[(u_s^{(i)})^2]. \end{aligned} \tag{2.16}$$

For discrete noise parameter, since φ_j only appears in $p(y_j^{(i)} | v_j^{(i)})$, we can similarly treat all other terms as constant. By substituting the discrete likelihood distribution with Eqn. (2.7), we have another specified objective of Eqn. (2.14)

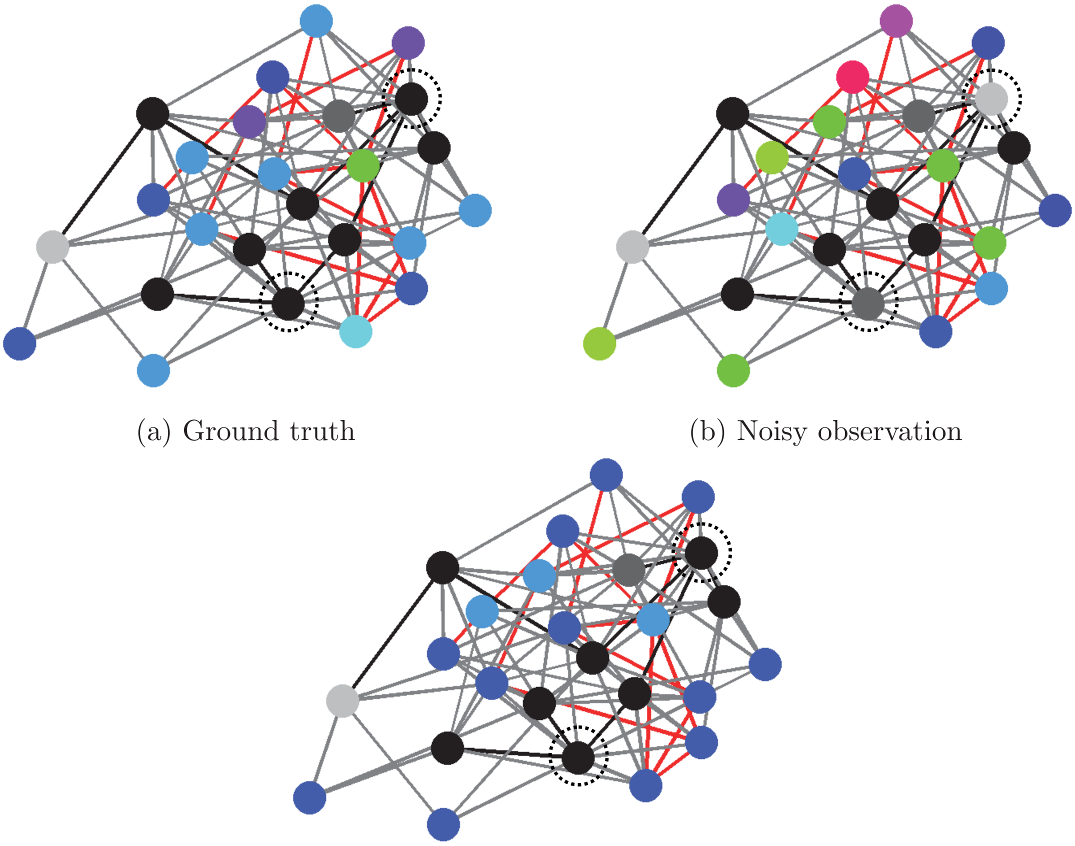
$$\begin{aligned}
\mathcal{Q}(\varphi_j) &\equiv \sum_i \sum_{v^{(i)}} \int_{u^{(i)}} q(u^{(i)}) q(v^{(i)}) \ln p(y_j^{(i)} | v_j^{(i)}) \\
&= \sum_i \sum_{v_j^{(i)}} q(v_j^{(i)}) \ln p(y_j^{(i)} | v_j^{(i)}) \\
&= \sum_i \sum_{v_j^{(i)}} q(v_j^{(i)}) \varphi_j(y_j^{(i)}, v_j^{(i)}) \\
&\quad - \sum_i \sum_{v_j^{(i)}} q(v_j^{(i)}) \ln \sum_l \exp\left(\varphi_j(l, v_j^{(i)})\right) \qquad (2.17)
\end{aligned}$$

Note that φ_j corresponds to a square table of size $L_j \times L_j$ and the degree of freedom for each column is $L_j - 1$ due to the normalization requirements. Now we specifically consider the element in a^{th} row and b^{th} column. In detail, take derivative of $\mathcal{Q}(\varphi_j)$ w.r.t. $\varphi_j(a, b)$ and set it to zero, we obtain

$$\begin{aligned}
&\sum_i q(v_j^{(i)} = b) \mathbb{I}(y_j^{(i)} = a) = \sum_i q(v_j^{(i)} = b) \frac{\exp(\varphi_j(a, b))}{\sum_l \exp(\varphi_j(l, b))} \\
\Rightarrow \quad & \frac{\exp(\varphi_j(a, b))}{\sum_l \exp(\varphi_j(l, b))} = \frac{\sum_i \mathbb{I}(y_j^{(i)} = a) q(v_j^{(i)} = b)}{\sum_i q(v_j^{(i)} = b)}. \qquad (2.18)
\end{aligned}$$

It is interesting that this equation seems to be indirectly related to an optimal $\varphi_j(a, b)$. However, $\frac{\exp(\varphi_j(a,b))}{\sum_l \exp(\varphi_j(l,b))}$ is exactly the desired probability $p(y_j = a | v_j = b)$. Upon this equation, the normalization issue is actually sidestepped.

So far, it seems that our derivation only considers noise. However, it is very straightforward to modify the proposed framework to handle missing values. Consider some of the observed variables of sample i are missing, say $x_m^{(i)}$ and $y_m^{(i)}$, we can simply delete those $p(x_m^{(i)} | u_m^{(i)})$ and $p(y_m^{(i)} | v_m^{(i)})$ terms, and keep all the other terms totally unchanged. Then the proposed variational inference procedure is modified accordingly. It is worth mentioning that our framework can resort to



(a) Ground truth

(b) Noisy observation

(c) Denoised result

Figure 2.3: The mixed-net graph used in our simulation contains 15 continuous (HSV-colored) and 10 discrete (grey-colored) nodes. The nodes are colored according to attribute values of a representative example. The three types of edges (continuous-continuous in red, discrete-discrete in black and continuous-discrete in light grey) are randomly chosen from all possible edges.

the generic RMF prior even when heavy missingness occurs. Thus, the simple deletion strategy is also applicable when the missing values become prevalent.

2.5 Experiments

2.5.1 Evaluation on Synthetic Data

We design a simulation study to show that mixed-attribute correlations can effectively help reduce noise effects and improve classification performance. Consider

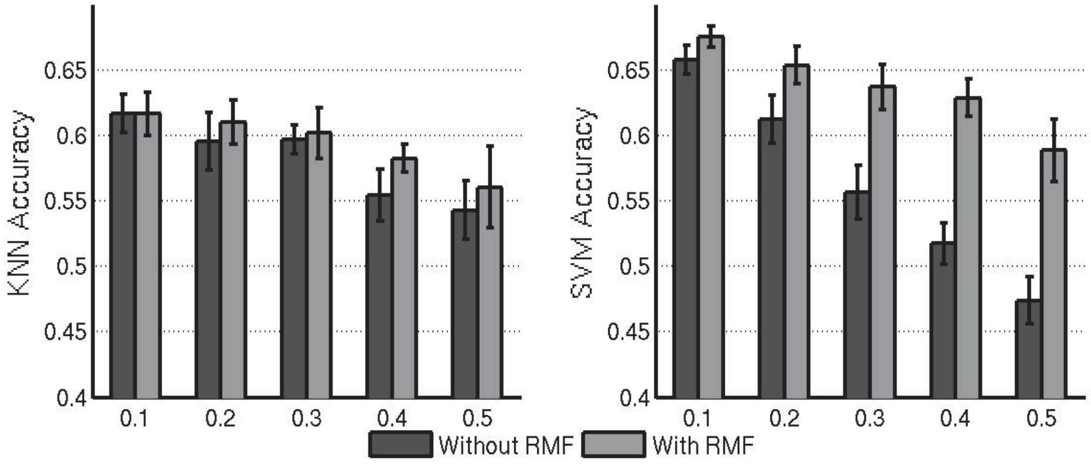


Figure 2.4: KNN (left plot) and SVM (right plot) classification accuracies of noisy (black) and denoised (light grey) data under different levels of random noise (the noise strength τ ranges from 0.1 to 0.5). Each bar represents the mean and standard deviation of 10 independent experiments.

a mixed-net graph consisting of 15 continuous and 10 discrete nodes with correlation parameters defined as below,

$$\begin{aligned}
&\mu_s = 1, \omega_{ss} = 1, \forall s \in \mathcal{V}_u, \omega_{st} = 4, \forall st \in \mathcal{E}_{uu};\\
&\rho_{sj} = [3\ 2\ 1], \forall sj \in \mathcal{E}_{uv};\\
&\phi_{jj} = 0, \forall j \in \mathcal{V}_v, \phi_{jk} = \begin{bmatrix} 1.5 & 0.5 & 0.5 \\ 0.5 & 1.5 & 0.5 \\ 0.5 & 0.5 & 1.5 \end{bmatrix}, \forall jk \in \mathcal{E}_{vv}.
\end{aligned}$$

In addition, we formulate two classes by adding two small but different random numbers ($\delta_1, \delta_2 \in [-0.5, 0.5]$) to all elements of ρ_{sj}. According to this setting, we generate 750 random examples for each class. Then we split all the examples into training and testing sets with a ratio of 2 : 1. The training set is utilized to train RMF model and KNN, SVM classifiers. And the testing set is used to generate noisy testing sets by injecting different levels of AWGN to continuous attributes and RFN to discrete attributes. In addition, the corruption strength is defined at five different percentages, i.e., $\tau = 0.1, 0.2, 0.3, 0.4, 0.5$. For continuous attributes, the noise standard deviations are $\sigma_s = \tau\check{\sigma}_s$, $s = 1, 2, \ldots, m$, with $\check{\sigma}_s$ being the signal standard deviations. For discrete attributes, the flipping probabilities are formulated as $p(y_j \neq a | v_j = a) = \tau$, $j = 1, 2, \ldots, n$.

Figure 2.3 illustrates the synthetic mixed-net graph structure and a representative example. We observe that the colors of these denoised continuous nodes are much closer to ground truth than noisy observation. In addition, the observed wrong state values of the discrete variables (in dotted circles) are also corrected after applying our inference algorithm. Besides the qualitative result, we also conduct quantitative classification experiment and summarize the results in Figure 2.4. According to the error bars, RMF improves the performance of classification significantly.

2.5.2 Evaluation on Real Data

In this section, we present experimental results on four real-world mixed-attribute datasets from the UCI machine learning repository [79], which are “Adult”, “Credit”, “Statlog-Australian” and “Statlog-German” as described in Table 2.1. The “Adult” dataset has already been split into train/test in approximately 2/3, 1/3 proportions. As for the “Credit”, “Statlog-Australian” and “Statlog-German” datasets, we simply select the first 2/3 proportion of all the instances as the training set and the remaining as the testing set.

Furthermore, to specifically consider the effect of all comparison methods on handling noise/missingness at testing stage, the experimental setting is clean training data versus corrupted testing data. The same methodology has also been widely employed in the literature, for example [34, 86, 111]. Consequently, all models and classifiers are built using clean training data and applied to handle corrupted testing data.

Except where no corruption is applied, each reported result is the average

Table 2.1: Datasets Summary. #Train./#Test. Inst. Stands for the Number of Training and Testing Instances Respectively. #Num./#Cat. Attr. Stands for the Number of Numerical and Categorical Attributes Respectively.

Datasets	#Train. Inst.	#Test. Inst.	#Num. Attr.	#Cat. Attr.	Testing Set Class Distrib.
Adult	32561	15060	6	8	[0.2457, **0.7543**]
Credit	460	220	6	9	[0.3909, **0.6091**]
Statlog-AU	460	230	6	8	[**0.5609**, 0.4391]
Statlog-GE	667	333	7	13	[**0.7117**, 0.2883]

classification accuracy over 10 independent experiments in which random noise and missingness are injected into the testing data. More importantly, all comparison methods are carried out on the same random noisy or incomplete testing data.

2.5.3 Data Denoising

For data denoising task, we employ the same noisy data generation strategy used in previous simulation study. More specifically, five different levels of noise strength ($\tau = 0.1, 0.2, 0.3, 0.4, 0.5$) are applied to all the four UCI datasets. Table 2.2 presents the classification accuracies of standard classifiers, before and after applying RMF denoising. As expected, the classification accuracy decreases as the noise strength increases compared to noise-free data classification. On the other hand, for most cases, the classification accuracies are effectively improved after RMF denoising. In addition, SVM classifier is more sensitive to noise than KNN classifier as the performance drops faster. In fact, SVM makes predictions using pre-trained fixed hyperplane weights while KNN is a lazy learner which can make adjustments for new instances.

2.5.4 Noisy Data Imputation

We further evaluate RMF's capability on the task of data imputation under noise. A little different from previous setting, the corrupted testing data are generated

Table 2.2: Classification Accuracies with/without Data Denoising.

τ	0	0.1	0.2	0.3	0.4	0.5	0	0.1	0.2	0.3	0.4	0.5
Method	Adult						Credit					
KNN	0.8248	**0.8174**	0.8063	0.7944	0.7821	0.7694	0.8227	**0.7727**	0.7191	0.6605	0.6105	0.5273
RMF+KNN		**0.8174**	**0.8083**	**0.7967**	**0.7865**	**0.7748**		0.7700	**0.7386**	**0.6936**	**0.6764**	**0.6455**
SVM	0.8467	0.8356	0.8243	0.8084	0.7951	0.7817	0.8636	**0.7895**	0.7200	0.6455	0.5900	0.4968
RMF+SVM		**0.8413**	**0.8317**	**0.8186**	**0.8053**	**0.7920**		**0.7895**	**0.7382**	**0.6859**	**0.6664**	**0.6336**
Method	Statlog-Australian						Statlog-German					
KNN	0.8783	0.8052	0.7274	0.6617	0.6148	0.5526	0.7417	**0.7189**	0.6895	0.6700	0.6703	0.6474
RMF+KNN		**0.8091**	**0.7613**	**0.7396**	**0.7074**	**0.6635**		0.7147	**0.6970**	**0.6880**	**0.6757**	**0.6655**
SVM	0.8478	0.7791	0.6948	0.6422	0.5752	0.4965	0.7688	0.7486	0.7204	0.6955	0.6778	0.6580
RMF+SVM		**0.7974**	**0.7661**	**0.7361**	**0.7078**	**0.6657**		**0.7523**	**0.7411**	**0.7210**	**0.7270**	**0.7096**

Table 2.3: Classification Accuracies with Noisy Data Imputation when $\tau = 0.2$.

ρ	0	0.1	0.3	0.5	0.7	0.9	0	0.1	0.3	0.5	0.7	0.9
Method	Adult						Credit					
KNNI+KNN	0.8063	0.8018	0.7814	0.7543	0.7075	0.3577	0.7191	0.7114	0.6859	0.6377	0.5673	0.4645
REMI+KNN	0.8063	**0.8043**	**0.7918**	**0.7683**	0.7333	0.6929	0.7191	0.7136	0.7045	0.6705	**0.6441**	0.6005
missF+KNN	0.8063	0.8033	0.7844	0.7605	0.7323	0.6940	0.7191	0.7205	0.7159	0.6586	0.6018	0.5068
RMFI+KNN	**0.8083**	0.7997	0.7794	0.7631	**0.7553**	**0.7543**	**0.7386**	**0.7245**	**0.7041**	**0.6732**	0.6355	**0.6150**
KNNI+SVM	0.8243	0.8144	0.7944	0.7703	0.7236	0.4947	0.7200	0.7095	0.6764	0.6300	0.5477	0.4555
REMI+SVM	0.8243	0.8190	**0.8088**	**0.7889**	0.7574	0.7032	0.7200	0.7150	0.6995	0.6668	0.6291	0.5327
missF+SVM	0.8243	0.8171	0.7974	0.7742	0.7460	0.7127	0.7200	0.7209	0.7168	0.6668	0.6064	0.4718
RMFI+SVM	**0.8317**	**0.8222**	0.7972	0.7743	**0.7589**	**0.7547**	**0.7382**	**0.7205**	**0.7091**	**0.6795**	**0.6377**	**0.6155**
Method	Statlog-Australian						Statlog-German					
KNNI+KNN	0.7274	0.7261	0.6952	0.6470	0.6000	0.5696	0.6895	0.6952	0.6913	0.6550	0.6057	0.5637
REMI+KNN	0.7274	0.7317	**0.7222**	0.6917	**0.6596**	**0.6000**	0.6895	**0.7030**	0.6874	**0.6784**	**0.6604**	0.5817
missF+KNN	0.7274	0.7400	0.7296	0.6978	0.6435	0.5348	0.6895	0.6970	0.6916	0.6706	0.6679	**0.6405**
RMFI+KNN	**0.7613**	**0.7635**	0.7217	**0.6926**	0.6470	0.5857	**0.6970**	0.6913	**0.6946**	0.6712	0.6465	0.6141
KNNI+SVM	0.6948	0.7022	0.6800	0.6474	0.5896	0.4917	0.7204	0.7282	0.7015	0.6895	0.5844	0.4147
REMI+SVM	0.6948	0.7070	0.7017	**0.7013**	**0.6600**	0.5770	0.7204	**0.7327**	0.6994	0.6976	0.6838	0.6793
missF+SVM	0.6948	0.7117	0.7087	0.6874	0.6430	0.5183	0.7204	0.7345	0.6955	0.6901	0.6808	0.6505
RMFI+SVM	**0.7661**	**0.7700**	**0.7317**	0.7009	0.6561	**0.5857**	**0.7411**	0.7312	**0.7144**	**0.7129**	**0.7072**	**0.7081**

by first injecting noise ($\tau = 0.2, 0.4$), then adding different levels of missingness. Missing completely at random (MCAR) strategy is employed to randomly annihilate a percentage ($\rho = 0.1, 0.3, 0.5, 0.7, 0.9$) of continuous and discrete attributes of each instance in the testing data. Tables 2.3 and 2.4 compare classification accuracies of standard classifiers utilizing KNN imputation (KNNI), regularized-EM imputation (REMI), random forest based imputation (missF) and the proposed RMF imputation (RMFI) techniques. According to the experimental results, RMFI obtained better performance with other imputation methods. Note that the proposed RMFI framework is capable of reducing noise effect during imputation, thus RMFI is very suitable for the noisy data imputation task.

Note that, KNNI imputes missing values with the mean/mode of K nearest

Table 2.4: Classification Accuracies with Noisy Data Imputation when $\tau = 0.4$.

ρ	0	0.1	0.3	0.5	0.7	0.9	0	0.1	0.3	0.5	0.7	0.9
Method	Adult						Credit					
KNNI+KNN	0.7821	0.7773	0.7612	0.7393	0.6954	0.3560	0.6105	0.5973	0.5668	0.5605	0.5332	0.4377
REMI+KNN	0.7821	0.7781	0.7675	0.7484	0.7178	0.6906	0.6105	0.6086	0.5782	0.5927	0.5959	0.6009
missF+KNN	0.7821	0.7779	0.7641	0.7480	0.7300	0.6704	0.6105	0.6168	0.6045	0.5918	0.5605	0.4545
RMFI+KNN	**0.7865**	**0.7800**	**0.7676**	**0.7589**	**0.7545**	**0.7543**	**0.6764**	**0.6655**	**0.6332**	**0.6255**	**0.6318**	**0.6123**
KNNI+SVM	0.7951	0.7869	0.7660	0.7441	0.7084	0.5013	0.5900	0.5809	0.5582	0.5450	0.5159	0.4332
REMI+SVM	0.7951	0.7925	**0.7817**	0.7639	0.7336	0.7001	0.5900	0.5864	0.5736	0.5864	0.5868	0.5191
missF+SVM	0.7951	0.7903	0.7744	0.7563	0.7371	0.6717	0.5900	0.5932	0.5923	0.5755	0.5568	0.4491
RMFI+SVM	**0.8053**	**0.7985**	0.7801	**0.7650**	**0.7574**	**0.7547**	**0.6664**	**0.6573**	**0.6259**	**0.6245**	**0.6305**	**0.6123**
Method	Statlog-Australian						Statlog-German					
KNNI+KNN	0.6148	0.6209	0.5839	0.5596	0.5587	0.5426	0.6703	0.6607	0.6462	0.6369	0.5994	0.5607
REMI+KNN	0.6148	0.6196	0.6052	0.6043	0.6152	**0.5709**	0.6703	0.6517	0.6468	0.6511	**0.6508**	0.5895
missF+KNN	0.6148	0.6291	0.6117	0.6074	0.6043	0.5083	0.6703	0.6514	0.6547	**0.6637**	0.6381	**0.6438**
RMFI+KNN	**0.7074**	**0.6900**	**0.6613**	**0.6496**	**0.6243**	0.5700	**0.6757**	**0.6682**	**0.6643**	0.6517	0.6303	0.6123
KNNI+SVM	0.5752	0.5848	0.5548	0.5383	0.5396	0.4904	0.6778	0.6793	0.6718	0.6465	0.5721	0.3955
REMI+SVM	0.5752	0.5935	0.5817	0.5935	0.6178	0.5504	0.6778	0.6763	0.6757	0.6637	0.6492	0.6580
missF+SVM	0.5752	0.5943	0.5917	0.5887	0.5970	0.4857	0.6778	0.6718	0.6793	0.6577	0.6450	0.6351
RMFI+SVM	**0.7078**	**0.6887**	**0.6657**	**0.6487**	**0.6252**	**0.5700**	**0.7270**	**0.7018**	**0.6979**	**0.7006**	**0.7057**	**0.7102**

neighbors for continuous/discrete attributes, while REMI is to impute missing values via a regularized expectation-maximization iterative procedure. In all our experiments, we employ the KNNI implementation, "knnimpute.m", from Matlab's Bioinformatics toolbox. For KNNI settings, we choose $K = 3$ and use weighted Euclidean distance measure, which is also suggested by the authors [124]. The REMI source code is available at the author's homepage[1], and the default setting is used. The R source code of missF[2] is called from Matlab via reading and writing "csv" files. We use the default setting for missF except for "Adult" dataset where $ntree = 1$ is employed for speed. Before applying the KNNI and REMI methods, we first transform those nominal attributes into dummy variables. Then the imputed testing data are post-processed to satisfy the constraint that the dummy vector of each nominal attribute should contain exactly one numerical value "1". It is worth mentioning that we have also tried other methodologies, such as specially impute nominal attributes with mode, but obtained no better results than the above one.

[1] http://climate-dynamics.org/software/#regem

[2] https://github.com/stekhoven/missForest

2.5.5 Noisy Data Imputation After Denoising

In this subsection, we further evaluate all the imputation algorithms under different levels of simulated noise (AWGN and RFN respectively). We choose to compare all the methods in this setting with all four datasets. The proposed RMFI framework is capable of reducing noise effect during imputation, thus RMFI is very suitable for the noisy data imputation task. To make a much fair comparison, a pre-step of RMF denoising is applied before all the other imputation methods are carried out. To give a complete comparison, we also include the results without applying RMF pre-denoising step as a reference point. Figures 2.5 - 2.12 present all the degeneration curves of classification accuracy versus number of missing attributes under different noise strength.

In general, a pair of solid and dashed lines through one shape (circle, lower triangle or diamond) represent the classification performance after imputation with or without RMF denoising. By and large, the former is slightly better than the latter. This observation validates the effectiveness of RMF for data denoising. Besides, all the degeneration curves in solid lines represent the classification performance after data restoration (whether in two stage as RMF plus imputation, or in one stage as RMFI). It is obvious that all these curves in one plot start from the same point which corresponds to the complete data case. For the six plots of one classifier (KNN or SVM) on one dataset, the starting point declines as the noise strength increases from 0 to 0.5. Under some specific level of noise strength, both KNN and SVM classification accuracies shrink as the number of missing features increases.

In particular, we shall mention five specific observations from all the comparison results. First of all, KNNI performs the worst for most of the cases, especially when the number of missing features goes beyond half of the total number. Secondly, on “Adult” dataset, RMFI is comparable with REMI and missF in both KNN and SVM classification accuracy when missingness is very light, while RMFI

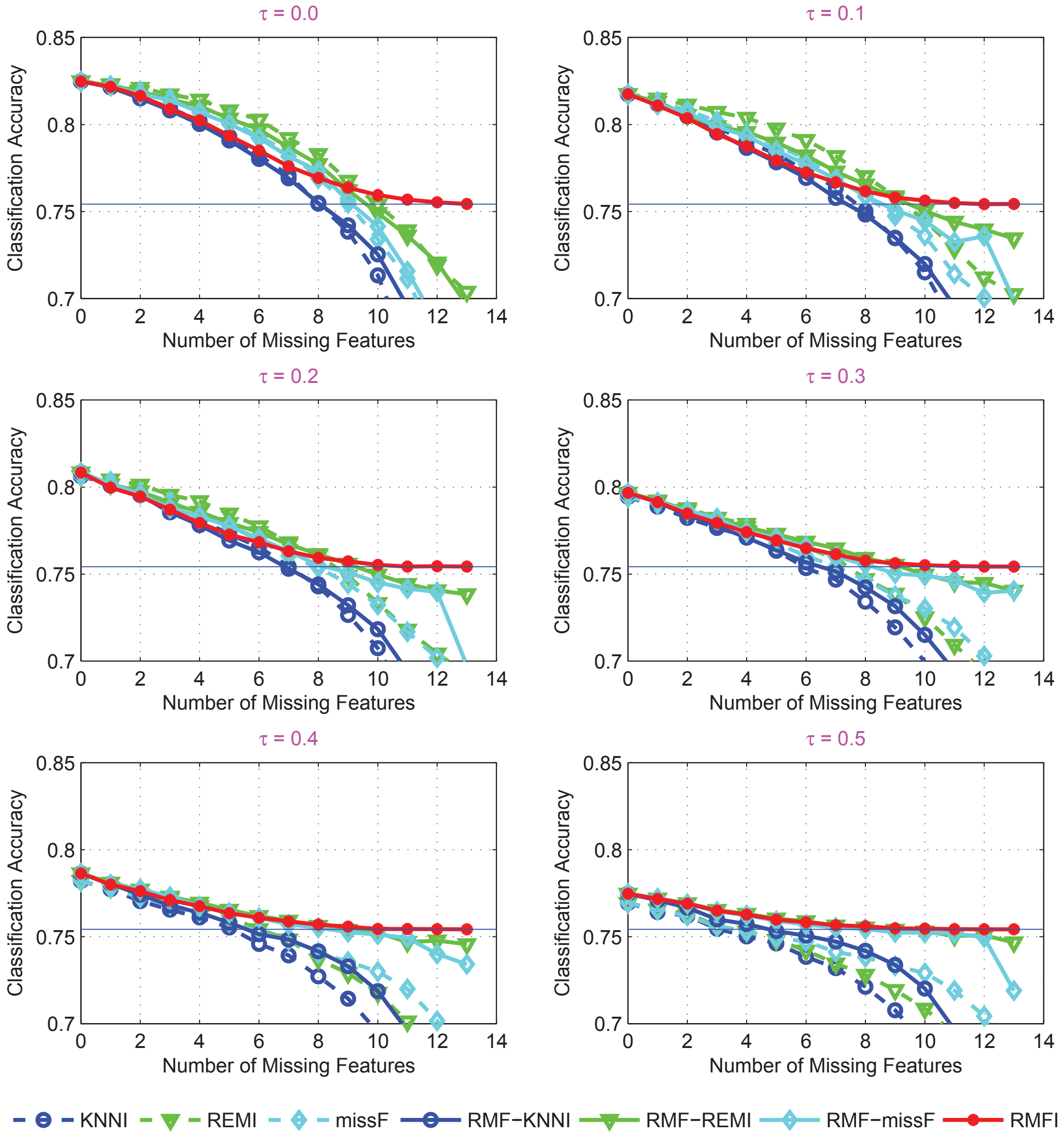


Figure 2.5: The degeneration curves of classification accuracy versus number of missing attributes on "Adult" dataset: Imputation + KNN Classifier

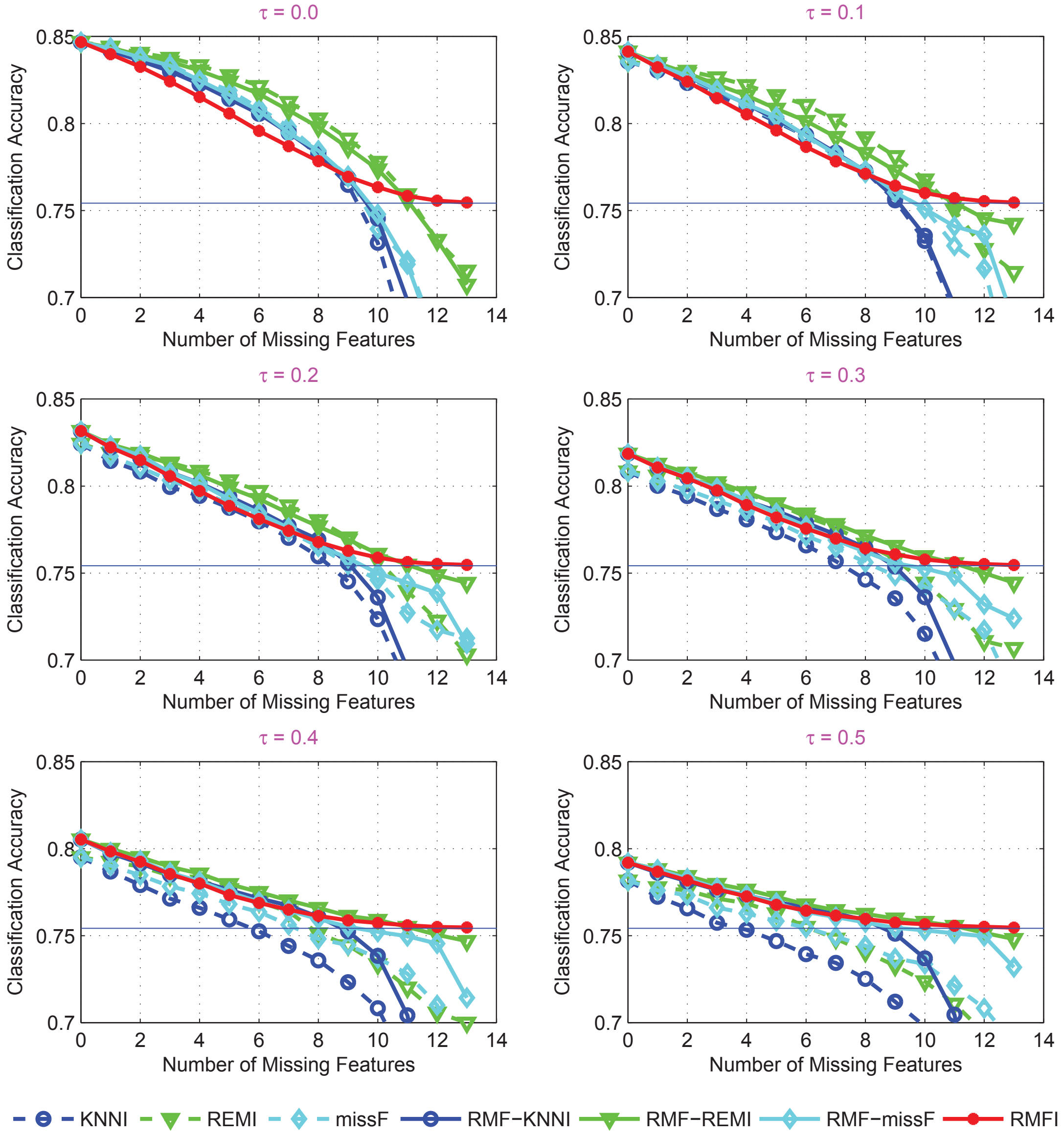


Figure 2.6: The degeneration curves of classification accuracy versus number of missing attributes on "Adult" dataset: Imputation + SVM Classifier

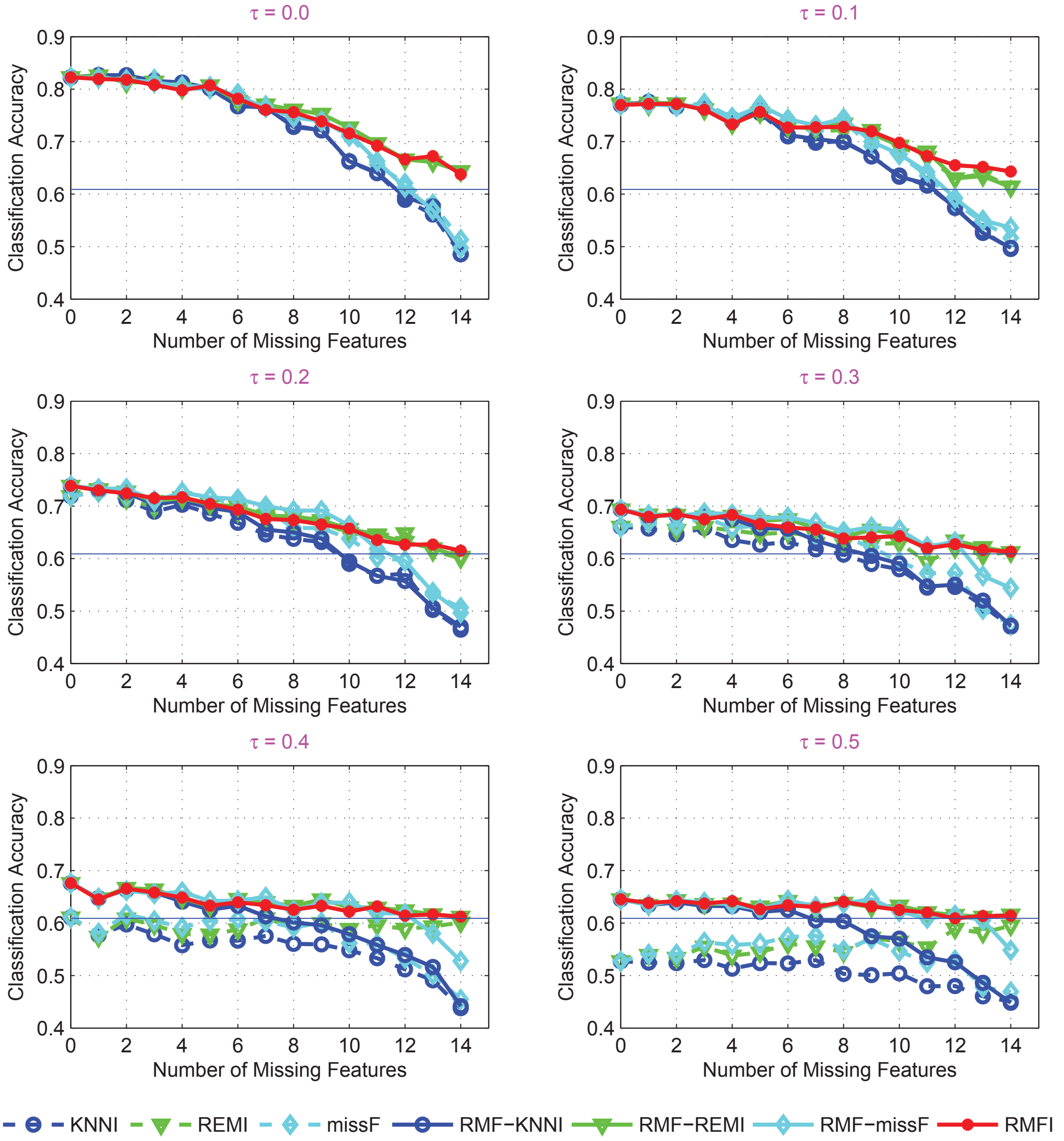


Figure 2.7: The degeneration curves of classification accuracy versus number of missing attributes on "Credit" dataset: Imputation + KNN Classifier

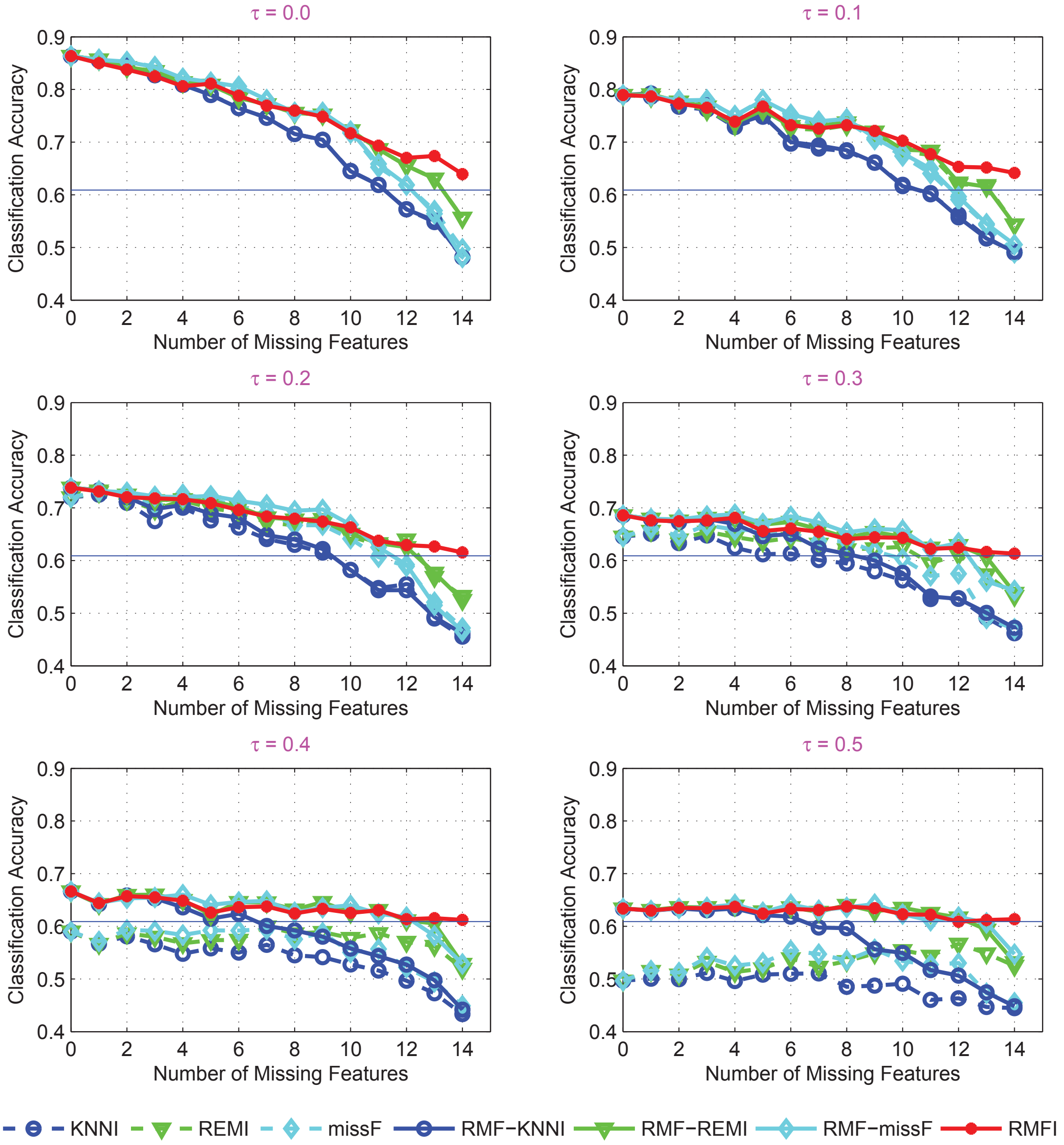


Figure 2.8: The degeneration curves of classification accuracy versus number of missing attributes on "Credit" dataset: Imputation + SVM Classifier

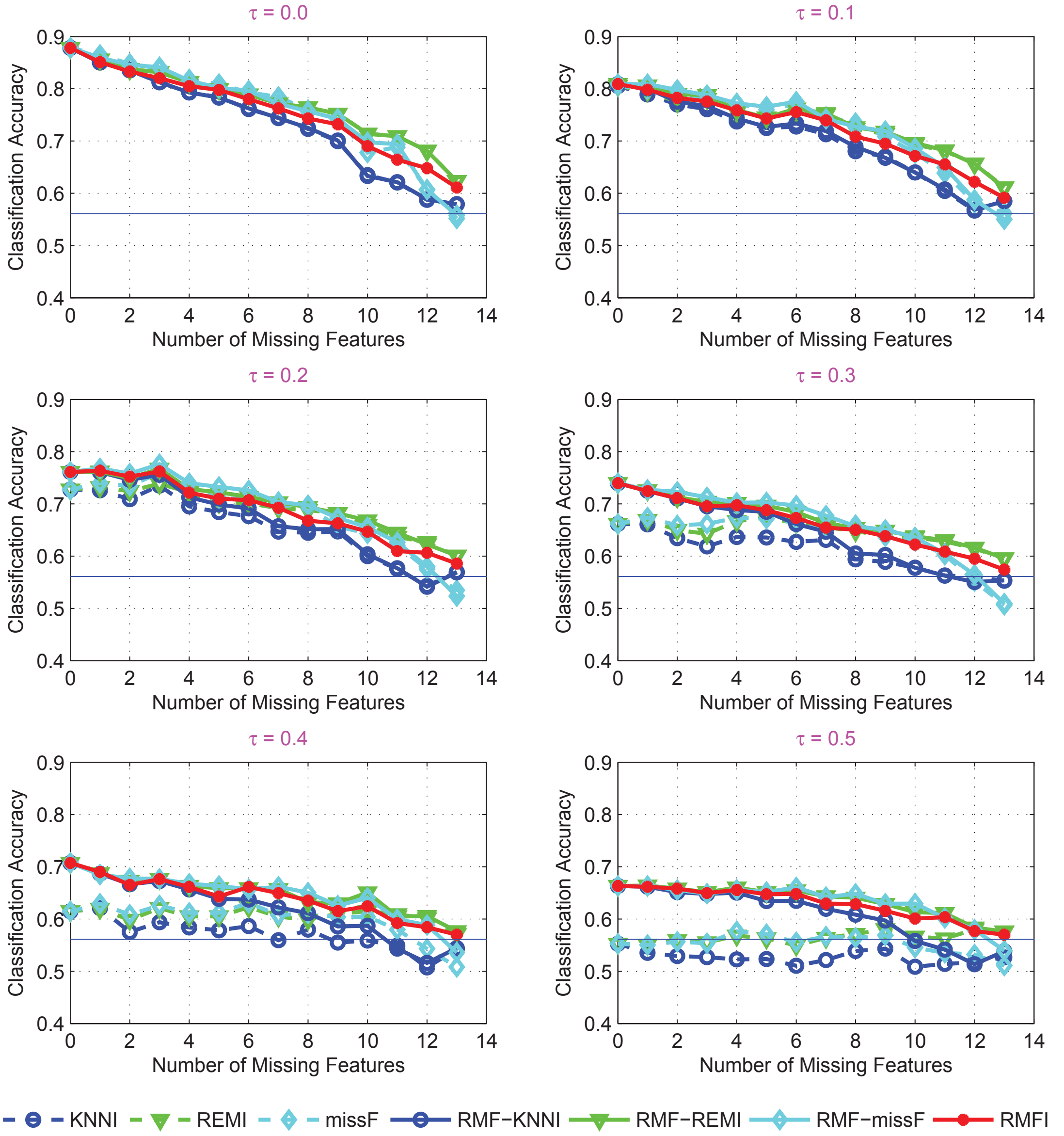


Figure 2.9: The degeneration curves of classification accuracy versus number of missing attributes on “Statlog-Australian” dataset: Imputation + KNN Classifier

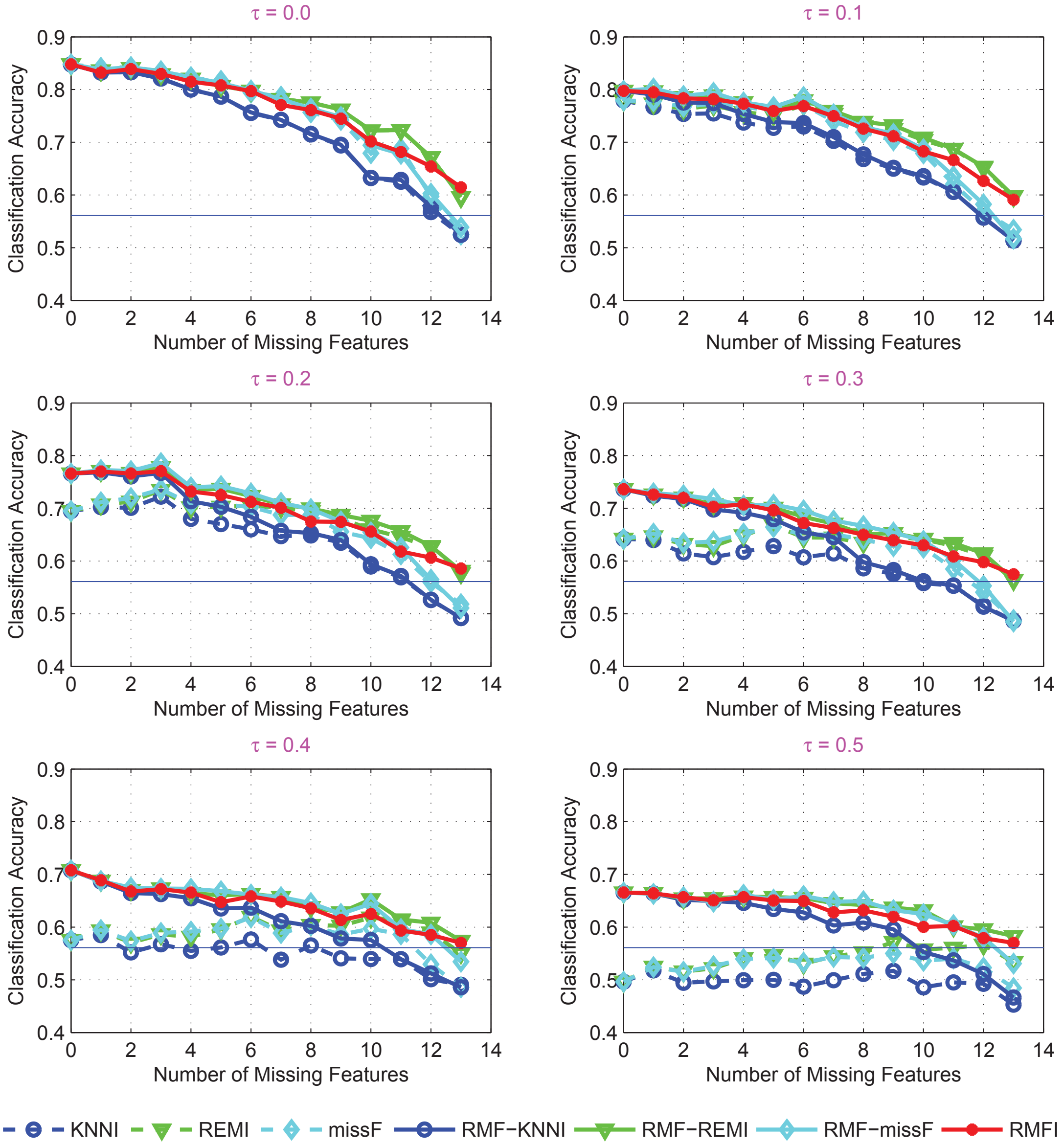


Figure 2.10: The degeneration curves of classification accuracy versus number of missing attributes on "Statlog-Australian" dataset: Imputation + SVM Classifier

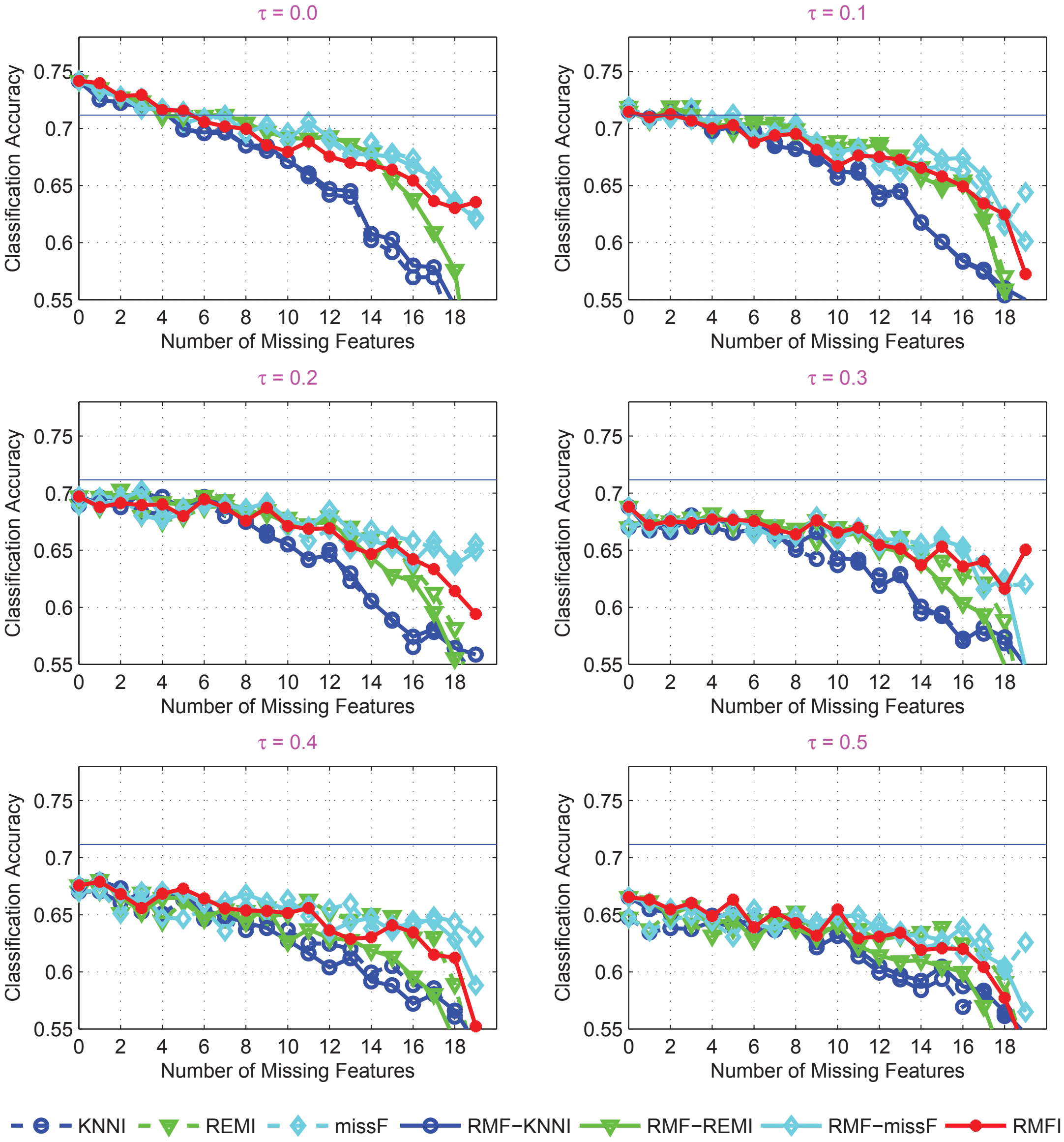


Figure 2.11: The degeneration curves of classification accuracy versus number of missing attributes on “Statlog-German” dataset: Imputation + KNN Classifier

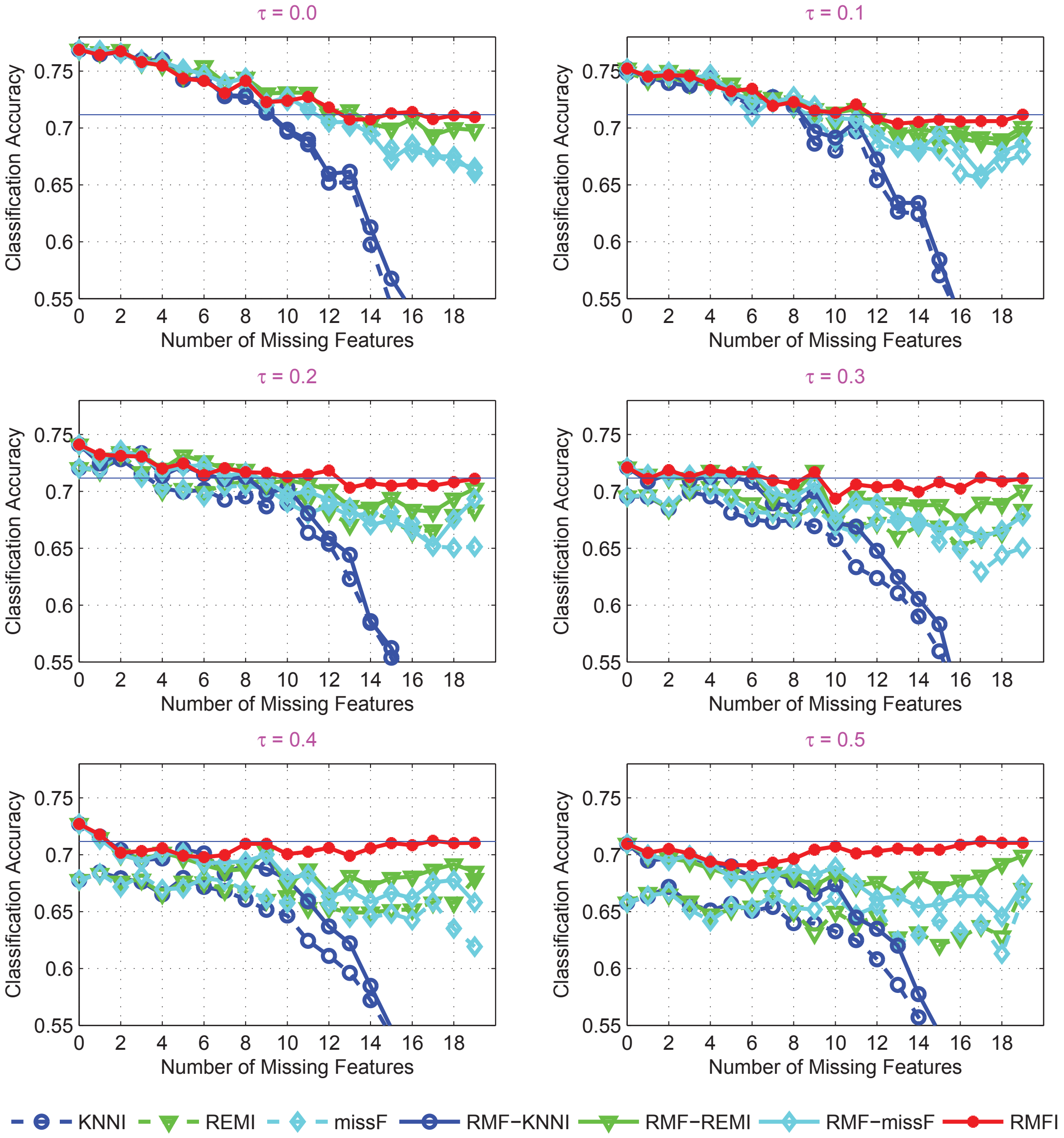


Figure 2.12: The degeneration curves of classification accuracy versus number of missing attributes on “Statlog-German” dataset: Imputation + SVM Classifier

surpasses REMI and missF when heavy missingness appears. This is reasonable since RMFI can still work in an average sense when the missingness is heavy. Thirdly, on “Credit” dataset, RMFI still works better than REMI and missF when a large missingness ratio occurs, while performs comparable with REMI and missF when the missingness is moderate. Fourthly, on “Statlog-Australian” dataset, though RMFI is slightly outperformed by REMI, the gap decreases as the noise strength gets heavier. Fifthly, on “Statlog-German’ dataset, RMFI is outperformed by missF with KNN classifier, but performs the best with SVM classifier. Consequently, we can claim that RMFI is more consistent than the other competing methods based on the above five observations.

2.6 Summary

Data restoration is common and critical for real-world data analysis practice. Although major problems, e.g, data denoising and imputation, have been widely studied in the literature, there still lacks a principled approach that is able to dress the generic data restoration problem. The proposed RMF model reduces this gap, by providing a principled approach to jointly handle data denoising and imputation within the probabilistic graphical model scope. An efficient inference algorithm for the RMF model was derived based on a structured variational approach. Empirical evaluations confirmed the effectiveness of RMF and showed its competitiveness by comparing with other data restoration methods.

Chapter 3

Elastic-Net Correlated Logistic Model for Multi-Label Classification

In this chapter, we present correlated logistic model (CorrLog) for multi-label classification. CorrLog extends conventional Logistic Regression model into multi-label cases, via explicitly modelling the pairwise correlation between labels. In addition, we propose to learn model parameters of CorrLog with Elastic Net regularization, which helps exploit the sparsity in feature selection and label correlations and thus further boost the performance of multi-label classification. CorrLog can be efficiently learned, though approximately, by regularized maximum pseudo likelihood estimation (MPLE), and it enjoys a satisfying generalization bound that is independent of the number of labels. We evaluate CorrLog's performance on two multi-label classification tasks. For music annotation, CorrLog achieves comparable results with the state-of-the-art performance on CAL-500 dataset. For multi-label image classification, CorrLog also performs competitively on benchmark datasets MULAN scene, MIT outdoor scene, PASCAL VOC 2007 and PASCAL VOC 2012, compared to the state-of-the-art multi-label classification algorithms.

3.1 Introduction

Multi-label classification extends conventional single label classification by allowing an instance to be assigned to multiple labels from a label set. It occurs naturally from a wide range of practical problems, such as document categorization, music annotation, image classification, and bioinformatics applications, where each instance can be simultaneously described by several class labels out of a candidate label set. Because of its great generality and wide applications, multi-label classification has received increasing attentions in recent years from machine learning, data mining, to computer vision communities, and developed rapidly with both algorithmic and theoretical achievements [28, 53, 102, 125, 152, 153].

The key feature of multi-label classification that makes it distinct from single-label classification is label correlation, without which classifiers can be trained independently for each individual label and multi-label classification degenerates to single-label classification. The correlation between different labels can be verified by calculating the statistics, e.g., χ^2 test and Pearson's correlation coefficient, of their distributions. According to [35, 36], there are two types of label correlations (or dependence), i.e., the conditional correlations and the unconditional correlations, wherein the former describes the label correlations conditioned on a given instance while the latter summarizes the global label correlations of only label distribution by marginalizing out the instance. From a classification point of view, modelling of label conditional correlations is preferable since they are directly related to prediction; however, proper utilization of unconditional correlations is also helpful, but in an average sense because of the marginalization.

In this chapter, we present correlated logistic model (CorrLog) to handle conditional label correlations in a more principle way. CorrLog enjoys several favourable properties: 1) built upon independent logistic regressions (ILRs), it offers an explicit way to model the pairwise (second order) label correlations; 2) by using the pseudo likelihood technique, the parameters of CorrLog can be learned approximately with a computational complexity linear with respect to label number; 3) the learning of CorrLog is stable, and the empirically learned model enjoys a generalization error bound that is independent of label number. In addition, the results presented here extend our previous study [10] in following aspects: 1) we introduce elastic net regularization to CorrLog, which facilitates the utilization of the sparsity in both feature selection and label correlations; 2) a learning algorithm for CorrLog based on soft thresholding is derived to handle the nonsmoothness of the elastic net regularization; 3) the proof of generalization bound is also extended for the new regularization; 4) we apply CorrLog to music annotation and multi-label image classification, and achieve competitive results with the state-of-the-art methods of their areas.

The rest of this chapter is organized as follows. Section 3.2 briefly reviews related methods for multi-label classification from the view of label dependence. Section 3.3 introduces the model CorrLog with elastic net regularization. Section 3.4 presents algorithms for learning CorrLog by regularized maximum pseudo likelihood estimation, and for prediction with CorrLog by message passing. Section 3.5 presents a generalization analysis of CorrLog based on the concept of algorithm stability. Sections 3.6 and 3.7 report results of empirical evaluations, including experiments on several benchmark datasets for music annotation and multi-label image classification.

3.2 Related Works

Quite a number of multi-label classification algorithms have been proposed in the past a few years, by exploiting either of unconditional or conditional label correlations [35, 36]. We give a brief review of the representative multi-label classification methods according to this taxonomy as below.

Methods exploiting unconditional label correlations effect in a global sense. A large class of multi-label classification algorithms that utilize unconditional label correlations are built upon label transformation. The main process is a three-phase procedure: (1) represent the original label vector [1] in an embedded subspace hoping that the dimensions of the embedded label vector (now maybe binary or continuous valued) are uncorrelated to each other; (2) perform single-label prediction independently in the embedded subspace; (3) recover the original label vector. Label transformation based methods include [147] which utilizes low-dimensional embedding and [53] and [156] which use random projections. Another strategy of using unconditional label correlations, e.g., used in the stacking method [28] and the "Curds" & "Whey" procedure [17], is first to predict each individual label independently and correct/adjust the prediction by proper post-processing. Algorithms are also proposed based on co-occurrence or structure information extracted from the label set, which include random k-label sets (RAKEL) [126], pruned problem transformation (PPT) [107], hierarchical binary relevance (HBR) [22] and hierarchy of multi-label classifiers (HOMER) [125]. Regression-based models, including reduced-rank regression and multitask learning, can also be used for multi-label classification, with an interpretation of utilizing unconditional label correlations [35].

Methods exploiting conditional label correlations perform in a local way. Multi-label classification algorithms in this category are diverse and often developed by specific heuristics. For example, multi-label k-nearest neighbour (MLKNN) [152] extends KNN to the multi-label situation, which applies maximum a posterior (MAP) label prediction by obtaining the prior label distribution within the k nearest neighbours of an instance. Instance-based logistic regression (IBLR) [28] is also a localized algorithm, which modifies logistic regression

[1] In general, a label vector is binary valued (denoted by either $\{0, 1\}$ or $\{-1, +1\}$) and each dimension decides the existence of that label. Note its difference from one-hot representation which corresponds to the multi-class case.

by using label information from the neighbourhood as features. Classifier chain (CC) [108], as well as its ensemble and probabilistic variants [29], incorporate label correlations into a chain of binary classifiers, where the prediction of a label uses previous labels as features. Channel coding based multi-label classification techniques such as principal label space transformation (PLST) [120] and maximum margin output coding (MMOC) [155] proposed to select codes that exploits conditional label correlations. Graphical models, e.g., conditional random fields (CRFs) [63], are also applied to multi-label classification, which provides a richer framework to handle conditional label correlations.

3.3 Correlated Logistic Model

We study the problem of learning a joint prediction $\mathbf{y} = d(\mathbf{x}) : \mathcal{X} \mapsto \mathcal{Y}$, where the instance space $\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^D\}$ and the label space $\mathcal{Y} = \{-1, 1\}^m$. By assuming the conditional independence among labels, we can model multi-label classification by a set of independent logistic regressions (ILRs). Specifically, the conditional probability $p_{lr}(\mathbf{y}|\mathbf{x})$ of ILRs is given by

$$\begin{aligned} p_{lr}(\mathbf{y}|\mathbf{x}) &= \prod_{i=1}^{m} p_{lr}(\mathbf{y}_i|\mathbf{x}) \\ &= \prod_{i=1}^{m} \frac{\exp\left(\mathbf{y}_i \beta_i^T \mathbf{x}\right)}{\exp\left(\beta_i^T \mathbf{x}\right) + \exp\left(-\beta_i^T \mathbf{x}\right)}, \end{aligned} \tag{3.1}$$

where $\beta_i \in \mathbb{R}^D$ is the coefficients for the i-th logistic regression (LR) in ILRs. For the convenience of expression, the bias of the standard LR is omitted here, which is equivalent to augmenting the feature of $\mathbf{x}$ with a constant.

Clearly, ILRs (3.1) enjoys several merits, such as, it can be learned efficiently, in particular with a linear computational complexity with respect to label number m, and its probabilistic formulation inherently helps deal with the imbalance of positive and negative examples for each label, which is a common problem encountered by multi-label classification. However, it ignores entirely the potential correlation among labels and thus tends to under-fit the true posterior $p_0(\mathbf{y}|\mathbf{x})$, especially when the label number m is large.

3.3.1 Correlated Logistic Regressions

CorrLog tries to extend ILRs with as small effort as possible, so that the correlation among labels is explicitly modelled while the advantages of ILRs can be also preserved. To achieve this, we propose to augment (3.1) with a simple function $q(\mathbf{y})$ and reformulate the posterior probability as

$$p(\mathbf{y}|\mathbf{x}) \propto p_{lr}(\mathbf{y}|\mathbf{x})q(\mathbf{y}). \tag{3.2}$$

As long as $q(\mathbf{y})$ cannot be decomposed into independent product terms for individual labels, it introduces label correlations into $p(\mathbf{y}|\mathbf{x})$. It is worth noticing that we assumed $q(\mathbf{y})$ to be independent of $\mathbf{x}$. Therefore, (3.2) models label correlations in an average sense. This is similar to the concept of "marginal correlations" in multi-label classification [35]. However, they are intrinsically different, because (3.2) integrate the correlation into the posterior probability, which directly aims at prediction. In addition, the idea used in (3.2) for correlation modelling is also distinct from the "Curds and Whey" procedure in [17] which corrects outputs of multivariate linear regression by reconsidering their correlations to the true responses.

In particular, we choose $q(\mathbf{y})$ to be the following quadratic form,

$$q(\mathbf{y}) = \exp\left\{\sum_{i<j} \alpha_{ij}\mathbf{y}_i\mathbf{y}_j\right\}. \tag{3.3}$$

It means that $\mathbf{y}_i$ and $\mathbf{y}_j$ are positively correlated given $\alpha_{ij} > 0$ and negatively correlated given $\alpha_{ij} < 0$. It is also possible to define α_{ij} as functions of $\mathbf{x}$, but this will drastically increase the number of model parameters, e.g., by $\mathcal{O}(m^2D)$ if linear functions are used.

By substituting (3.3) into (3.2), we obtain the conditional probability for CorrLog

$$p(\mathbf{y}|\mathbf{x};\Theta) \propto \exp\left\{\sum_{i=1}^{m} \mathbf{y}_i \beta_i^T \mathbf{x} + \sum_{i<j} \alpha_{ij} \mathbf{y}_i \mathbf{y}_j\right\}, \tag{3.4}$$

where the model parameter $\Theta = \{\boldsymbol{\beta}, \boldsymbol{\alpha}\}$ contains $\boldsymbol{\beta} = [\beta_1, ..., \beta_m]$ and $\boldsymbol{\alpha} = [\alpha_{12}, ..., \alpha_{(m-1)m}]^T$. It can be seen that CorrLog is a simple modification of (3.1), by using a quadratic term to adjust the joint prediction, so that hidden label correlations can be exploited. In addition, CorrLog is closely related to popular statistical models for joint modelling of binary variables. For example, conditional on $\mathbf{x}$, (3.4) is exactly an Ising model [105] for $\mathbf{y}$. It can also be treated as a special instance of CRFs [63], by defining features $\phi_i(\mathbf{x}, \mathbf{y}) = \mathbf{y}_i \mathbf{x}$ and $\psi_{ij}(\mathbf{y}) = \mathbf{y}_i \mathbf{y}_j$. Moreover, classical model multivariate probit (MP) [2] also models pairwise correlations in $\mathbf{y}$. However, it utilizes Gaussian latent variables for correlation modelling, which is essentially different from CorrLog.

3.3.2 Elastic Net Regularization

Given a set of training data $\mathcal{D} = \{\mathbf{x}^{(l)}, \mathbf{y}^{(l)} : 1 \leq l \leq n\}$, CorrLog can be learned by regularized maximum log likelihood estimation (MLE), i.e.,

$$\widehat{\Theta} = \arg\min_{\Theta} \mathcal{L}(\Theta) + R(\Theta), \tag{3.5}$$

where $\mathcal{L}(\Theta)$ is the negative log likelihood

$$\mathcal{L}(\Theta) = -\frac{1}{n} \sum_{l=1}^{n} \log p(\mathbf{y}^{(l)}|\mathbf{x}^{(l)};\Theta), \tag{3.6}$$

and $R(\Theta)$ is a properly chosen regularization.

A possible choice for $R(\Theta)$ is the ℓ_2 regularizer,

$$R_2(\Theta; \lambda_1, \lambda_2) = \lambda_1 \sum_{i=1}^{m} \|\beta_i\|_2^2 + \lambda_2 \sum_{i<j} |\alpha_{ij}|^2, \tag{3.7}$$

with λ_1, $\lambda_2 > 0$ being the weighting parameters. The ℓ_2 regularization enjoys the merits of computational flexibility and learning stability. However, it is unable to exploit any sparsity that can be possessed by the problem at hand. For example, for multi-label classification, it is likely that the prediction of each label $\mathbf{y}_i$ only depends on a subset of the D features of $\mathbf{x}$, which implies the sparsity of β_i. Besides, $\boldsymbol{\alpha}$ can also be sparse since not all labels in $\mathbf{y}$ are correlated to each other. ℓ_1 regularizer is another choice for $\mathcal{R}(\Theta)$, especially regarding model sparsity. Nevertheless, it has been noticed by several studies that ℓ_1 regularized algorithms are inherently unstable, that is, a slight change of the training data set can lead to substantially different prediction models. Based on above consideration, we propose to use the elastic net regularizer [159], which is a combination of ℓ_2 and ℓ_1 regularizers and inherits their individual advantages, i.e., learning stability and model sparsity,

$$\begin{aligned} R_{en}(\Theta; \lambda_1, \lambda_2, \epsilon) = \lambda_1 &\sum_{i=1}^{m} (\|\beta_i\|_2^2 + \epsilon\|\beta_i\|_1) \\ + \lambda_2 &\sum_{i<j} (|\alpha_{ij}|^2 + \epsilon|\alpha_{ij}|), \end{aligned} \tag{3.8}$$

where $\epsilon \geq 0$ controls the trade-off between the ℓ_1 regularization and the ℓ_2 regularization, and large ϵ encourages a high level of sparsity.

3.4 Algorithms

In this section, we derive algorithms for learning and prediction with CorrLog. The exponentially large size of the label space $\mathcal{Y} = \{-1, 1\}^m$ makes exact algorithms for CorrLog computationally intractable, since the conditional probability (3.4) needs to be normalized by the log-partition function

$$A(\Theta) = \log \sum_{y \in \mathcal{Y}} \exp \left\{ \sum_{i=1}^{m} \mathbf{y}_i \beta_i^T \mathbf{x} + \sum_{i<j} \alpha_{ij} \mathbf{y}_i \mathbf{y}_j \right\}, \tag{3.9}$$

which involves a summation over an exponential number of terms. Thus, we turn to approximate learning and prediction algorithms, by exploiting the pseudo likelihood and the message passing techniques.

3.4.1 Approximate Learning via Pseudo Likelihood

Maximum pseudo likelihood estimation (MPLE) [7] provides an alternative approach for estimating model parameters, especially when the partition function of the likelihood cannot be evaluated efficiently. It was developed in the field of spatial dependence analysis and has been widely applied to the estimation of various statistical models, from the Ising model [105] to the CRFs [119]. Here, we apply MPLE to the learning of parameter Θ in CorrLog.

The pseudo likelihood of the model over m jointly distributed random variables is defined as the product of the conditional probability of each individual random variables conditioned on all the rest ones. For CorrLog (3.4), its pseudo likelihood is given by

$$\widetilde{p}(\mathbf{y}|\mathbf{x};\Theta) = \prod_{i=1}^{m} p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta), \tag{3.10}$$

where $\mathbf{y}_{-i} = [\mathbf{y}_1, ..., \mathbf{y}_{i-1}, \mathbf{y}_{i+1}, ..., \mathbf{y}_m]$ and the conditional probability $p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta)$ can be directly obtained from (3.4),

$$p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta) = \frac{1}{1 + \exp\left\{-2\mathbf{y}_i \left(\beta_i^T \mathbf{x} + \sum_{j=i+1}^{m} \alpha_{ij} \mathbf{y}_j + \sum_{j=1}^{i-1} \alpha_{ji} \mathbf{y}_j\right)\right\}}. \tag{3.11}$$

Accordingly, the negative log pseudo likelihood over the training data $\mathcal{D}$ is given by

$$\widetilde{\mathcal{L}}(\Theta) = -\frac{1}{n}\sum_{l=1}^{n}\sum_{i=1}^{m}\log p(\mathbf{y}_i^{(l)}|\mathbf{y}_{-i}^{(l)},\mathbf{x}^{(l)};\Theta). \tag{3.12}$$

To this end, the optimal model parameter $\widetilde{\Theta} = \{\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}\}$ of CorrLog can be learned approximately by the elastic net regularized MPLE,

$$\begin{aligned}\widetilde{\Theta} &= \arg\min_{\Theta} \widetilde{\mathcal{L}}_r(\Theta) \\ &= \arg\min_{\Theta} \widetilde{\mathcal{L}}(\Theta) + R_{en}(\Theta;\lambda_1,\lambda_2,\epsilon).\end{aligned} \tag{3.13}$$

where λ_1, λ_2 and ϵ are tuning parameters.

A First-Order Method by Soft Thresholding: Problem (3.13) is a convex optimization problem, thanks to the convexity of the logarithmic loss function and the elastic net regularization, and thus a unique optimal solution. However, the elastic net regularization is non-smooth due to the ℓ_1 norm regularizer, which makes direct gradient based algorithm inapplicable. The main idea of our algorithm for solving (3.13) is to divide the objective function into smooth and non-smooth parts, and then apply the soft thresholding technique to deal with the non-smoothness.

Denoting by $J_s(\Theta)$ the smooth part of $\widetilde{\mathcal{L}}_r(\Theta)$, i.e.,

$$J_s(\Theta) = \widetilde{\mathcal{L}}(\Theta) + \lambda_1\sum_{i=1}^{m}\|\beta_i\|_2^2 + \lambda_2\sum_{i<j}|\alpha_{ij}|^2, \tag{3.14}$$

its gradient ∇J_s at the k-th iteration $\Theta^{(k)} = \{\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)}\}$ is given by

$$\begin{cases} \nabla J_{s\beta_i}(\Theta^{(k)}) = \frac{1}{n}\sum\limits_{l=1}^{n}\xi_{li}\mathbf{x}^{(l)} + 2\lambda_1\beta_i^{(k)} \\ \nabla J_{s\alpha_{ij}}(\Theta^{(k)}) = \frac{1}{n}\sum\limits_{l=1}^{n}\left(\xi_{li}\mathbf{y}_j^{(l)} + \xi_{lj}\mathbf{y}_i^{(l)}\right) + 2\lambda_2\alpha_{ij}^{(k)} \end{cases} \tag{3.15}$$

with

$$\xi_{li} = \frac{-2\mathbf{y}_i^{(l)}}{1 + \exp\left\{2\mathbf{y}_i^{(l)} \left(\beta_i^{(k)T}\mathbf{x}^{(l)} + \sum_{j=i+1}^{m} \alpha_{ij}^{(k)}\mathbf{y}_j^{(l)} + \sum_{j=1}^{i-1} \alpha_{ji}^{(k)}\mathbf{y}_j^{(l)}\right)\right\}}. \tag{3.16}$$

Then, a surrogate $J(\Theta)$ of the objective function $\widetilde{\mathcal{L}}_r(\Theta)$ in (3.13) can be obtained by using $\nabla J_s(\Theta^{(k)})$, i.e.,

$$\begin{aligned} &J(\Theta;\Theta^{(k)}) = J_s(\Theta^{(k)}) \\ &+ \sum_{i=1}^{m} \langle \nabla J_{s\beta_i}(\Theta^{(k)}), \beta_i - \beta_i^{(k)} \rangle + \frac{1}{2\eta}\|\beta_i - \beta_i^{(k)}\|_2^2 + \lambda_1\epsilon\|\beta_i\|_1 \\ &+ \sum_{i<j} \langle \nabla J_{s\alpha_{ij}}(\Theta^{(k)}), \alpha_{ij} - \alpha_{ij}^{(k)} \rangle + \frac{1}{2\eta}(\alpha_{ij} - \alpha_{ij}^{(k)})^2 + \lambda_2\epsilon|\alpha_{ij}|. \end{aligned} \tag{3.17}$$

The parameter η in (4.25) servers a similar role to the variable updating step size in gradient descent methods, and it is set such that $1/\eta$ is larger than the Lipschitz constant of $\nabla J_s(\Theta^{(k)})$. For such η, it can be shown that $J(\Theta) \geq \widetilde{\mathcal{L}}_r(\Theta)$ and $J(\Theta^{(k)}) = \widetilde{\mathcal{L}}_r(\Theta^{(k)})$. Therefore, the update of Θ can be realized by the minimization

$$\Theta^{(k+1)} = \arg\min_{\Theta} J(\Theta;\Theta^{(k)}), \tag{3.18}$$

which is solved by the soft thresholding function $\mathcal{S}(\cdot)$, i.e.,

$$\begin{cases} \beta_i^{(k+1)} = \mathcal{S}(\beta_i^{(k)} - \eta\nabla J_{s\beta_i}(\Theta^{(k)}); \lambda_1\epsilon) \\ \alpha_{ij}^{(k+1)} = \mathcal{S}(\alpha_{ij}^{(k)} - \eta\nabla J_{s\alpha_{ij}}(\Theta^{(k)}); \lambda_2\epsilon), \end{cases} \tag{3.19}$$

Algorithm 1 Learning CorrLog by Maximum Pseudo Likelihood Estimation with Elastic Net Regularization

```
**Input:** Training data $\mathcal{D}$, initialization $\boldsymbol{\beta}^{(0)} = \mathbf{0}$, $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$, and learning rate $\eta$, where $1/\eta$ is set larger than the Lipschitz constant of $\nabla J_s(\Theta)$ (4.25).
**Output:** Model parameters $\widetilde{\Theta} = (\widetilde{\boldsymbol{\beta}}^{(t)}, \widetilde{\boldsymbol{\alpha}}^{(t)})$.
repeat
  Calculating the gradient of $J_S(\Theta)$ at $\Theta^{(k)} = (\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)})$ by using (4.24);
  Updating $\Theta^{(k+1)} = (\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})$ by using soft thresholding (4.27);
    $k = k + 1$
until Converged
```

where

$$\mathcal{S}(u;\rho) = \begin{cases} u - 0.5\rho, & \text{if } u > 0.5\rho \\ u + 0.5\rho, & \text{if } u < -0.5\rho \\ 0, & \text{otherwise.} \end{cases} \tag{3.20}$$

Iteratively applying (4.27) until convergence provides a first-order method for solving (3.13). Algorithm 1 presents the pseudo code for this procedure.

Remark 1 From the above derivation, especially equations (4.24) and (4.27), the computational complexity of our learning algorithm is linear with respect to the label number m. Therefore, learning CorrLog is no more expensive than learning m independent logistic regressions, which makes CorrLog scalable to the case of large label numbers.

Remark 2 It is possible to further speed up the learning algorithm. In particular, Algorithm 1 can be modified to have the optimal convergence rate in the sense of Nemirovsky and Yudin [96], i.e., $\mathcal{O}(1/k^2)$ wherein k is the number of iterations. However, its convergence is usually as slow as in standard gradient descent methods. Actually, we only need to replace the current variable $\Theta^{(k)}$ in the surrogate (4.25) by a weighted combination of the variables from previous iterations. As such modification is a direct application of the fast iterative shrinkage thresholding, [4], we do not present the details here but leave readers to the reference.

3.4.2 Joint Prediction by Message Passing

For multi-label classification, as the labels are not independent in general, the prediction task is actually a joint maximum a posterior (MAP) estimation over $p(\mathbf{y}|\mathbf{x})$. In the case of CorrLog, suppose the model parameter $\widetilde{\Theta}$ is learned by the regularized MPLE from the last subsection, the prediction of $\widehat{\mathbf{y}}$ for a new instance $\mathbf{x}$ can be obtained by

$$\begin{aligned} \widehat{\mathbf{y}} &= \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x};\widetilde{\Theta}) \\ &= \arg\max_{\mathbf{y}\in\mathcal{Y}} \exp\left\{\sum_{i=1}^{m} \mathbf{y}_i \widetilde{\beta}_i^T \mathbf{x} + \sum_{i<j} \widetilde{\alpha}_{ij} \mathbf{y}_i \mathbf{y}_j\right\}. \end{aligned} \tag{3.21}$$

We use the belief propagation (BP) to solve (3.21) [12]. Specifically, we run the max-product algorithm with uniformly initialized messages and an early stopping criterion with 50 iterations. Since the graphical model defined by $\boldsymbol{\alpha}$ in (3.21) has loops, we cannot guarantee the convergence of the algorithm. However, we found that it works well on all experiments in this study.

3.5 Generalization Analysis

An important issue in designing a machine learning algorithm is generalization, i.e., how the algorithm will perform on the test data compared to on the training data. In the section, we present a generalization analysis for CorrLog, by using the concept of algorithmic stability [14]. Our analysis follows two steps. First, we show that the learning of CorrLog by MPLE is stable, i.e., the learned model parameter $\widetilde{\Theta}$ does not vary much given a slight change of the training data set $\mathcal{D}$. Then, we prove that the generalization error of CorrLog can be bounded by the empirical error, plus a term related to the stability but independent of the label

Table 3.1: Summary of important notations for generalization analysis.

Notation	Description
$\mathcal{D} = \{\mathbf{x}^{(l)}, \mathbf{y}^{(l)}\}$	training dataset with n examples, $1 \leq l \leq n$
$\mathcal{D}^k$	modified training data set by replacing the k-th example of $\mathcal{D}$ with an independent example
$\mathcal{D}^{\backslash k}$	modified training data set by discarding the k-th example of $\mathcal{D}$
$\widetilde{\mathcal{L}}(\Theta)$	negative log pseudo likelihood over training dataset $\mathcal{D}^k$
$\widetilde{\mathcal{L}}_r(\Theta)$	regularized negative log pseudo likelihood over training dataset $\mathcal{D}^{\backslash k}$
$R_{en}(\Theta; \lambda_1, \lambda_2, \epsilon)$	elastic net regularization with weights λ_1, λ_2 and parameter ϵ
$\Theta = \{\boldsymbol{\beta}, \boldsymbol{\alpha}\}$	model parameters of CorrLog
$\widetilde{\Theta} = \{\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}\}$	empirical learned model parameters by maximum pseudo likelihood estimation over $\mathcal{D}$
$\widetilde{\Theta}^k = \{\widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\alpha}}^k\}$	empirical learned model parameters over $\mathcal{D}^k$
$\widetilde{\Theta}^{\backslash k} = \{\widetilde{\boldsymbol{\beta}}^{\backslash k}, \widetilde{\boldsymbol{\alpha}}^{\backslash k}\}$	empirical learned model parameters over $\mathcal{D}^{\backslash k}$
$\widetilde{\mathcal{R}}(\widetilde{\Theta})$	empirical error of the empirical model $\widetilde{\Theta}$ over training set $\mathcal{D}$
$\mathcal{R}(\widetilde{\Theta})$	generalization error of the empirical model $\widetilde{\Theta}$

number m. To ease the presentation of generalization analysis, we summarize the important notations in Table 3.1.

3.5.1 The Stability of MPLE

The stability of a learning algorithm indicates how much the learned model changes according to a small change of the training data set. Denote by $\mathcal{D}^k$ a modified training data set the same with $\mathcal{D}$ but replacing the k-th training example $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ by another independent example $(\mathbf{x}', \mathbf{y}')$. Suppose $\widetilde{\Theta}$ and $\widetilde{\Theta}^k$ are the model parameters learned by MPLE (3.13) on $\mathcal{D}$ and $\mathcal{D}^k$, respectively. We intend to show that the difference between these two models, defined as

$$\|\widetilde{\Theta}^k - \widetilde{\Theta}\| \triangleq \sum_{i=1}^{m} \|\widetilde{\boldsymbol{\beta}}_i^k - \widetilde{\boldsymbol{\beta}}_i\| + \sum_{i<j} |\widetilde{\boldsymbol{\alpha}}_{ij}^k - \widetilde{\boldsymbol{\alpha}}_{ij}|, \ \forall\ 1 \leq k \leq n, \tag{3.22}$$

is bounded by an order of $\mathcal{O}(1/n)$, so that the learning is stable for large n.

First, we need the following auxiliary model $\widetilde{\Theta}^{\backslash k} = \{\widetilde{\boldsymbol{\beta}}^{\backslash k}, \widetilde{\boldsymbol{\alpha}}^{\backslash k}\}$ learned on $\mathcal{D}^{\backslash k}$, which is the same with $\mathcal{D}$ but without the k-th example

$$\widetilde{\Theta}^{\backslash k} = \arg\min_{\Theta} \widetilde{\mathcal{L}}^{\backslash k}(\Theta) + \mathcal{R}_{en}(\Theta; \lambda_1, \lambda_2, \epsilon), \tag{3.23}$$

where

$$\widetilde{\mathcal{L}}^{\backslash k}(\Theta) = -\frac{1}{n} \sum_{l \neq k} \sum_{i=1}^{m} \log p(\mathbf{y}_i^{(l)} | \mathbf{y}_{-i}^{(l)}, \mathbf{x}^{(l)}; \Theta). \tag{3.24}$$

The following Lemma provides an upper bound of the difference $\widetilde{\mathcal{L}}_r(\widetilde{\Theta}^{\backslash k}) - \widetilde{\mathcal{L}}_r(\widetilde{\Theta})$.

Lemma 1. *Given $\widetilde{\mathcal{L}}_r(\cdot)$ and $\widetilde{\Theta}$ defined in (3.13), and $\widetilde{\Theta}^{\backslash k}$ defined in (3.23), it* *holds for* $\forall 1 \le k \le n$,

$$\widetilde{\mathcal{L}}_r(\widetilde{\Theta}^{\backslash k}) - \widetilde{\mathcal{L}}_r(\widetilde{\Theta}) \le \\ \frac{1}{n}\left(\sum_{i=1}^{m} \log p(\mathbf{y}_i^{(k)}|\mathbf{y}_{-i}^{(k)}, \mathbf{x}^{(k)}; \widetilde{\Theta}^{\backslash k}) - \sum_{i=1}^{m} \log p(\mathbf{y}_i^{(k)}|\mathbf{y}_{-i}^{(k)}, \mathbf{x}^{(k)}; \widetilde{\Theta})\right) \tag{3.25}$$

Proof. Denote by RHS the righthand side of (3.25), we have

$$\text{RHS} = \left(\widetilde{\mathcal{L}}_r(\widetilde{\Theta}^{\backslash k}) - \widetilde{\mathcal{L}}_r^{\backslash k}(\widetilde{\Theta}^{\backslash k})\right) - \left(\widetilde{\mathcal{L}}_r(\widetilde{\Theta}) - \widetilde{\mathcal{L}}_r^{\backslash k}(\widetilde{\Theta})\right).$$

Furthermore, the definition of $\widetilde{\Theta}^{\backslash k}$ implies $\widetilde{\mathcal{L}}_r^{\backslash k}(\widetilde{\Theta}^{\backslash k}) \le \widetilde{\mathcal{L}}_r^{\backslash k}(\widetilde{\Theta})$. Combining these two we have (3.25). This completes the proof. □

Next, we show a lower bound of the difference $\widetilde{\mathcal{L}}_r(\widetilde{\Theta}^{\backslash k}) - \widetilde{\mathcal{L}}_r(\widetilde{\Theta})$.

Lemma 2. *Given* $\widetilde{\mathcal{L}}_r(\cdot)$ *and* $\widetilde{\Theta}$ *defined in (3.13), and* $\widetilde{\Theta}^{\backslash k}$ *defined in (3.23), it holds for* $\forall 1 \le k \le n$,

$$\widetilde{\mathcal{L}}_r(\widetilde{\Theta}^{\backslash k}) - \widetilde{\mathcal{L}}_r(\widetilde{\Theta}) \ge \lambda_1 \|\widetilde{\boldsymbol{\beta}}^{\backslash k} - \widetilde{\boldsymbol{\beta}}\|^2 + \lambda_2 \|\widetilde{\boldsymbol{\alpha}}^{\backslash k} - \widetilde{\boldsymbol{\alpha}}\|^2. \tag{3.26}$$

Proof. We define the following function

$$f(\Theta) = \widetilde{\mathcal{L}}_r(\Theta) - \lambda_1 \|\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}\|^2 - \lambda_2 \|\boldsymbol{\alpha} - \widetilde{\boldsymbol{\alpha}}\|^2.$$

Then, for (3.26), it is sufficient to show that $f(\widetilde{\Theta}^{\backslash k}) \ge f(\widetilde{\Theta})$. By using (3.13), we have

$$f(\Theta) = \widetilde{\mathcal{L}}(\Theta) + 2\lambda_1 \sum_{i=1}^{m} \beta_i^T \widetilde{\boldsymbol{\beta}}_i + 2\lambda_2 \sum_{i<j} \alpha_{ij}\widetilde{\alpha}_{ij} \\ + \lambda_1 \epsilon \sum_{i=1}^{m} \|\beta_i\|_1 + \lambda_2 \epsilon \sum_{i<j} |\alpha_{ij}|. \tag{3.27}$$

It is straightforward to verify that $f(\Theta)$ and $\widetilde{\mathcal{L}}_r(\Theta)$ in (3.13) have the same subgradient at $\widetilde{\Theta}$, i.e.,

$$\partial f(\widetilde{\Theta}) = \partial \widetilde{\mathcal{L}}_r(\widetilde{\Theta}). \tag{3.28}$$

Since $\widetilde{\Theta}$ minimizes $\widetilde{\mathcal{L}}_r(\Theta)$, we have $\mathbf{0} \in \partial \widetilde{\mathcal{L}}_r(\widetilde{\Theta})$ and thus $\mathbf{0} \in \partial f(\widetilde{\Theta})$, which implies $\widetilde{\Theta}$ also minimizes $f(\Theta)$. Therefore $f(\widetilde{\Theta}) \leq f(\widetilde{\Theta}^{\setminus k})$. □

In addition, by checking the Lipschitz continuous property of $\log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta)$, we have the following Lemma 3.

Lemma 3. *Given $\widetilde{\Theta}$ defined in (3.13) and $\widetilde{\Theta}^{\setminus k}$ defined in (3.23), it holds for $\forall\ (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ and $\forall 1 \leq k \leq n$*

$$\begin{aligned} &\Big|\sum_{i=1}^{m} \log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \widetilde{\Theta}) - \sum_{i=1}^{m} \log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \widetilde{\Theta}^{\setminus k})\Big| \\ &\leq 2\sum_{i=1}^{m} \|\widetilde{\beta}_i - \widetilde{\beta}_i^{\setminus k}\| + 4\sum_{i<j} |\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^{\setminus k}|. \end{aligned} \tag{3.29}$$

Proof. First, we have

$$\|\partial \log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta)/\partial \beta_i\| \leq 2\|\mathbf{x}\| \leq 2,$$

and

$$|\partial \log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta)/\partial \alpha_{ij}| \leq 4|\mathbf{y}_i\mathbf{y}_j| = 4.$$

That is $\log p(\mathbf{y}_i|\mathbf{y}_{-i}, \mathbf{x}; \Theta)$ is Lipschitz continuous with respect to β_i and α_{ij}, with constant 2 and 4, respectively. Therefore, (3.29) holds. □

By combining the above three Lemmas, we have the following Theorem 1 that shows the stability of CorrLog.

Theorem 1. *Given model parameters $\widetilde{\Theta} = \{\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}\}$ and $\widetilde{\Theta}^k = \{\widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\alpha}}^k\}$ learned on training datasets $\mathcal{D}$ and $\mathcal{D}^k$, respectively, both by (3.13), it holds that*

$$\sum_{i=1}^{m} \|\widetilde{\boldsymbol{\beta}}_i^k - \widetilde{\boldsymbol{\beta}}_i\| + \sum_{i<j} |\widetilde{\boldsymbol{\alpha}}_{ij}^k - \widetilde{\boldsymbol{\alpha}}_{ij}| \leq \frac{16}{\min(\lambda_1, \lambda_2)n}. \tag{3.30}$$

Proof. By combining (3.25), (3.26) and (3.29), we have

$$\begin{aligned} &\|\widetilde{\boldsymbol{\beta}}^{\setminus k} - \widetilde{\boldsymbol{\beta}}\|^2 + \|\widetilde{\boldsymbol{\alpha}}^{\setminus k} - \widetilde{\boldsymbol{\alpha}}\|^2 \leq \\ &\frac{4}{\min(\lambda_1, \lambda_2)n} \left(\sum_{i=1}^{m} \|\widetilde{\beta}_i - \widetilde{\beta}_i^{\setminus k}\| + \sum_{i<j} |\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^{\setminus k}| \right). \end{aligned} \tag{3.31}$$

Further, by using

$$\begin{aligned} &\|\widetilde{\boldsymbol{\beta}}^{\setminus k} - \widetilde{\boldsymbol{\beta}}\|^2 + \|\widetilde{\boldsymbol{\alpha}}^{\setminus k} - \widetilde{\boldsymbol{\alpha}}\|^2 \geq \\ &\qquad \frac{1}{2} \left(\sum_{i=1}^{m} \|\widetilde{\beta}_i - \widetilde{\beta}_i^{\setminus k}\| + \sum_{i<j} |\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^{\setminus k}| \right)^2 \end{aligned} \tag{3.32}$$

we have

$$\sum_{i=1}^{m} \|\widetilde{\beta}_i - \widetilde{\beta}_i^{\setminus k}\| + \sum_{i<j} |\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^{\setminus k}| \leq \frac{8}{\min(\lambda_1, \lambda_2)n} \tag{3.33}$$

Since $\mathcal{D}^k$ and $\mathcal{D}^{\setminus k}$ differ from each other with only the k-th training example, the same argument gives

$$\sum_{i=1}^{m} \|\widetilde{\beta}_i^k - \widetilde{\beta}_i^{\setminus k}\| + \sum_{i<j} |\widetilde{\alpha}_{ij}^k - \widetilde{\alpha}_{ij}^{\setminus k}| \leq \frac{8}{\min(\lambda_1, \lambda_2)n}. \tag{3.34}$$

Then, (3.30) is obtained immediately. This completes the proof. □

3.5.2 Generalization Bound

We first define a loss function to measure the generalization error. Considering that CorrLog predicts labels by MAP estimation, we define the loss function by using the log probability

$$\ell(\mathbf{x},\mathbf{y};\Theta) = \begin{cases} 1, & f(\mathbf{x},\mathbf{y},\Theta) < 0 \\ 1 - f(\mathbf{x},\mathbf{y},\Theta)/\gamma, & 0 \leq f(\mathbf{x},\mathbf{y},\Theta) < \gamma \\ 0, & f(\mathbf{x},\mathbf{y},\Theta) \geq \gamma, \end{cases} \tag{3.35}$$

where the constant $\gamma > 0$ and

$$\begin{aligned} f(\mathbf{x},\mathbf{y},\Theta) &= \log p(\mathbf{y}|\mathbf{x};\Theta) - \max_{\mathbf{y}'\neq\mathbf{y}} \log p(\mathbf{y}'|\mathbf{x};\Theta) \\ &= \left(\sum_{i=1}^{m} \mathbf{y}_i \beta_i^T \mathbf{x} + \sum_{i<j} \alpha_{ij} \mathbf{y}_i \mathbf{y}_j \right) \\ &\quad - \max_{\mathbf{y}'\neq\mathbf{y}} \left(\sum_{i=1}^{m} \mathbf{y}'_i \beta_i^T \mathbf{x} + \sum_{i<j} \alpha_{ij} \mathbf{y}'_i \mathbf{y}'_j \right). \end{aligned} \tag{3.36}$$

The loss function (3.35) is defined analogously to the loss function used in binary classification, where $f(\mathbf{x},\mathbf{y},\Theta)$ is replaced with the margin $y\mathbf{w}^T\mathbf{x}$ if a linear classifier $\mathbf{w}$ is used. Besides, (3.35) gives a 0 loss only if all dimensions of $\mathbf{y}$ are correctly predicted, which emphasizes the joint prediction in multi-label classification. By using this loss function, the generalization error and the empirical error are given by

$$\mathcal{R}(\widetilde{\Theta}) = \mathbb{E}_{\mathbf{xy}} \ell(\mathbf{x},\mathbf{y};\widetilde{\Theta}), \tag{3.37}$$

and

$$\widetilde{\mathcal{R}}(\widetilde{\Theta}) = \frac{1}{n} \sum_{l=1}^{n} \ell(\mathbf{x}^{(l)},\mathbf{y}^{(l)};\widetilde{\Theta}). \tag{3.38}$$

According to [14], an exponential bound exists for $\mathcal{R}(\widetilde{\Theta})$ if CorrLog has a uniform stability with respect to the loss function (3.35). The following Theorem

2 shows this condition holds.

Theorem 2. *Given model parameters* $\widetilde{\Theta} = \{\widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}\}$ *and* $\widetilde{\Theta}^k = \{\widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\alpha}}^k\}$ *learned on training datasets* $\mathcal{D}$ *and* $\mathcal{D}^k$*, respectively, both by (3.13), it holds for* $\forall(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$*,*

$$|\ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}) - \ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}^k)| \leq \frac{32}{\gamma \min(\lambda_1, \lambda_2) n}. \tag{3.39}$$

Proof. First, we have the following inequality from (3.35)

$$\gamma|\ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}) - \ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}^k)| \leq |f(\mathbf{x}, \mathbf{y}, \widetilde{\Theta}) - f(\mathbf{x}, \mathbf{y}, \widetilde{\Theta}^k)| \tag{3.40}$$

Then, by introducing notation

$$A(\mathbf{x}, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \mathbf{y}_i \beta_i^T \mathbf{x} + \sum_{i<j} \alpha_{ij} \mathbf{y}_i \mathbf{y}_j, \tag{3.41}$$

and rewriting

$$f(\mathbf{x}, \mathbf{y}, \Theta) = A(\mathbf{x}, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\alpha}) - \max_{\mathbf{y}' \neq \mathbf{y}} A(\mathbf{x}, \mathbf{y}', \boldsymbol{\beta}, \boldsymbol{\alpha}), \tag{3.42}$$

we have

$$\begin{aligned} \gamma|\ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}) - \ell(\mathbf{x}, \mathbf{y}; \widetilde{\Theta}^k)| \leq & \left|A(\mathbf{x}, \mathbf{y}, \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}) - A(\mathbf{x}, \mathbf{y}, \widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\alpha}}^k)\right| \\ & + |\max_{\mathbf{y}' \neq \mathbf{y}} A(\mathbf{x}, \mathbf{y}', \widetilde{\boldsymbol{\beta}}, \widetilde{\boldsymbol{\alpha}}) - \max_{\mathbf{y}' \neq \mathbf{y}} A(\mathbf{x}, \mathbf{y}', \widetilde{\boldsymbol{\beta}}^k, \widetilde{\boldsymbol{\alpha}}^k)|. \end{aligned} \tag{3.43}$$

Due to the fact that for any functions $h_1(u)$ and $h_2(u)$ it holds[1]

$$|\max_u h_1(u) - \max_u h_2(u)| \leq \max_u |h_1(u) - h_2(u)|, \tag{3.44}$$

[1] Suppose $u_1^\star$ and $u_2^\star$ maximize $h_1(u)$ and $h_2(u)$ respectively, and without loss of generality $h_1(u_1^\star) \geq h_2(u_2^\star)$, we have $|h_1(u_1^\star) - h_2(u_2^\star)| = h_1(u_1^\star) - h_2(u_2^\star) \leq h_1(u_1^\star) - h_2(u_1^\star) \leq \max_u |h_1(u) - h_2(u)|$.

we have

$$
\begin{aligned}
&\gamma|\ell(\mathbf{x},\mathbf{y};\widetilde{\Theta}) - \ell(\mathbf{x},\mathbf{y};\widetilde{\Theta}^k)| \\
&\leq \left|A(\mathbf{x},\mathbf{y},\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\alpha}}) - A(\mathbf{x},\mathbf{y},\widetilde{\boldsymbol{\beta}}^k,\widetilde{\boldsymbol{\alpha}}^k)\right| \\
&\qquad\qquad + \max_{\mathbf{y}'\neq\mathbf{y}} \left|A(\mathbf{x},\mathbf{y}',\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\alpha}}) - A(\mathbf{x},\mathbf{y}',\widetilde{\boldsymbol{\beta}}^k,\widetilde{\boldsymbol{\alpha}}^k)\right| \\
&\leq 2\max_{\mathbf{y}} \left(\sum_{i=1}^m |\mathbf{y}_i(\widetilde{\beta}_i - \widetilde{\beta}_i^k)^T\mathbf{x}| + \sum_{i<j} |(\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^k)\mathbf{y}_i\mathbf{y}_j|\right) \\
&\leq 2\left(\sum_{i=1}^m \|\widetilde{\beta}_i - \widetilde{\beta}_i^k\| + 2\sum_{i<j} |\widetilde{\alpha}_{ij} - \widetilde{\alpha}_{ij}^k|\right). \qquad (3.45)
\end{aligned}
$$

Then, the proof is completed by applying Theorem 1. □

Now, we are ready to present the main theorem on the generalization ability of CorrLog.

Theorem 3. *Given the model parameter* $\widetilde{\Theta}$ *learned by (3.13), with i.i.d. training data* $\mathcal{D} = \{(\mathbf{x}^{(l)},\mathbf{y}^{(l)}) \in \mathcal{X}\times\mathcal{Y}, l = 1,2,...,n\}$ *and regularization parameters* λ_1, λ_2, *it holds with at least probability* $1-\delta$,

$$
\begin{aligned}
\mathcal{R}(\widetilde{\Theta}) \leq \widetilde{\mathcal{R}}(\widetilde{\Theta}) &+ \frac{32}{\gamma\min(\lambda_1,\lambda_2)n} \\
&+ \left(\frac{64}{\gamma\min(\lambda_1,\lambda_2)} + 1\right)\sqrt{\frac{\log 1/\delta}{2n}}. \qquad (3.46)
\end{aligned}
$$

Proof. Given Theorem 2, the generalization bound (3.46) is a direct result of Theorem 12 in [14] (Please refer to the reference for details). □

Remark 3 A notable observation from Theorem 3 is that the generalization bound (3.46) of CorrLog is independent of the label number m. Therefore, CorrLog is preferable for multi-label classification with a large number of labels, for which the generalization error still can be bounded with high confidence.

Remark 4 While the learning of CorrLog (3.13) utilizes the elastic net regularization $R_{en}(\Theta;\lambda_1,\lambda_2,\epsilon)$, where ϵ is the weighting parameter on the ℓ_1 regularization

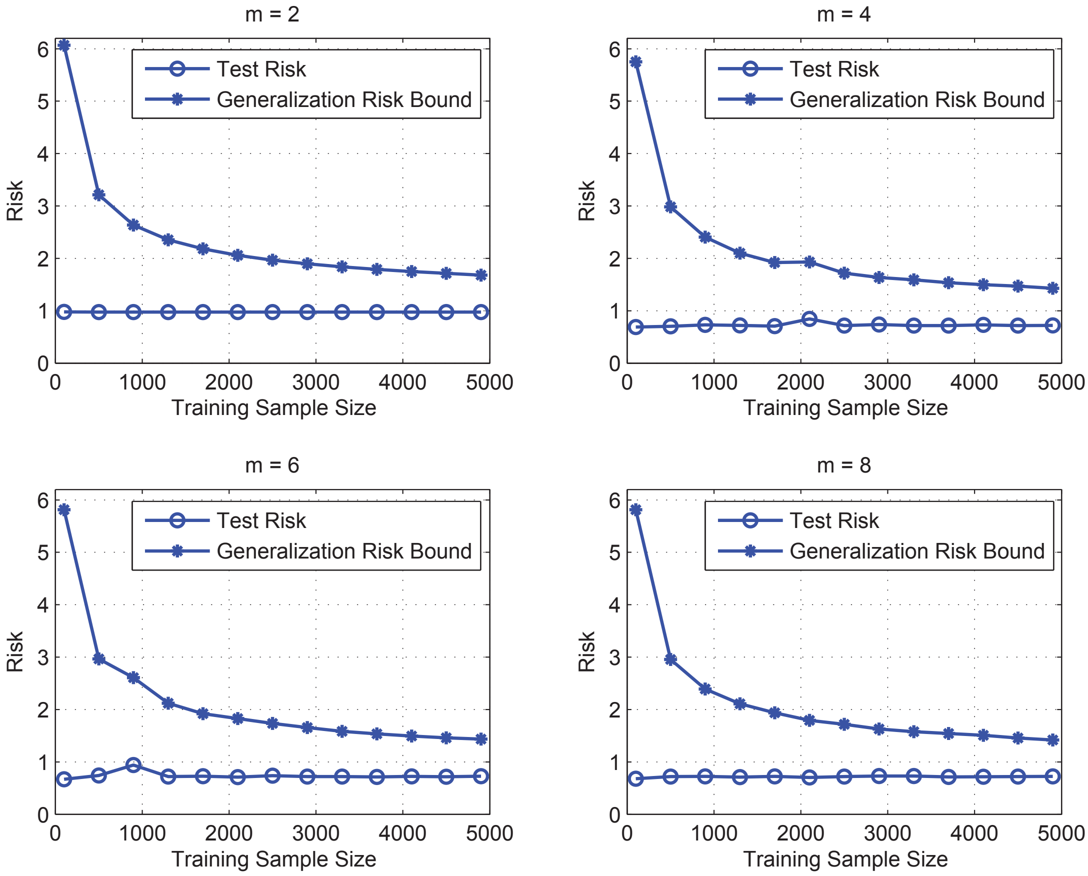


Figure 3.1: Empirical evaluation of the generalization bound of CorrLog with different number of labels.

to encourage sparsity, the generalization bound (3.46) is independent of the parameter ϵ. The reason is that ℓ_1 regularization does not lead to stable learning algorithms [140], and only the ℓ_2 regularization in $R_{en}(\Theta; \lambda_1, \lambda_2, \epsilon)$ contributes to the stability of CorrLog.

3.5.3 Empirical Evaluation

We design a simple simulation study to demonstrate the generalization capability of CorrLog. Consider a m-label classification problem on a 2-D plane, where each instance $\mathbf{x}$ is sampled uniformly from the unit disc $\|\mathbf{x}\| \leq 1$ and the corresponding labels $\mathbf{y} = [\mathbf{y}_1, ..., \mathbf{y}_m]$ are defined by

$$\mathbf{y}_i = \begin{cases} \text{sign}(\eta_1^T \tilde{\mathbf{x}}), & i = 1, \\ \text{OR}\left(\mathbf{y}_{i-1}, \text{sign}(\eta_{i-1}^T \tilde{\mathbf{x}})\right), & i = 2, ..., m, \end{cases}$$

Note that $\eta_1, ..., \eta_m$ are defined by using the first m vectors from a predefined pool[1] and the augmented feature is $\tilde{\mathbf{x}} = [\mathbf{x}^T, 1]^T$. The sign$(\cdot)$ function takes value 1 or -1, and the OR$(\cdot, \cdot)$ operation outputs 1 if either of its input is 1. The definition of $\mathbf{y}$ makes the m labels correlated. Then, we use different number of training samples, from 100 to 5000, to learn model parameter of CorrLog, and use 5000 test samples to estimate the test risk. The generalization risk is composed of the sample error and the approximation error, with a confidence set to 0.9 and the best model's risk estimated from 10 000 samples. The model parameter, used in both the empirical learning and calculating the sample error, is selected by the 10-fold cross validation on the training set. Figure 3.1 presents the simulation results. With the increase of the number of training samples, the generalization bound of CorrLog approaches to the test risk and the decrease rate is roughly $\sqrt{n}$ which is consistent with (3.46). In addition, the curves of four different number of labels are much similar to each other, which confirms the observation from Theorem 3 that the generalization bound is independent of the label number m.

3.6 Experiments: Music Annotation and Retrieval

In this section, we apply CorrLog to music annotation and retrieval. Given a song with a few relevant labels or tags, e.g. pop, male vocal and happy, we want to predict confidence values that accurately estimate the strength of the

[1] The predefined pool includes $\eta_1 = (1, 1, -0.5)$, $\eta_2 = (-1, 1, -0.5)$, $\eta_3 = (1, -1, -0.5)$, $\eta_4 = (-1, -1, -0.5)$, $\eta_5 = (1, 2, -0.5)$, $\eta_6 = (1, -2, -0.5)$, $\eta_7 = (-1, 2, -0.5)$, and $\eta_8 = (-1, -2, -0.5)$.

association between the labels and audio contents. These confidence values can be used to rank the tags by relevance, and this is the music annotation task. In the music retrieval task, we rank the songs according to their relevance to a specific query tag. We show that CorrLog provides promising performance for both music annotation and retrieval compared with the state-of-the-arts.

Table 3.2: Experimental results for top 97 popular tags. CBA stands for Codeword Bernoulli Average (CBA) [51], GMM for Gaussian Mixture Models [128], DirMix for Dirichlet Mixture model [92].

Model	Precision	Recall	F-score	P3	P5	P10	MAP	AROC
CorrLog	**0.452**	0.229	**0.314**	0.517	**0.524**	**0.487**	**0.462**	**0.717**
CBA	0.361	0.212	0.267	0.463	0.458	0.440	0.425	0.691
GMM	0.405	0.202	0.269	0.456	0.455	0.441	0.433	0.698
Context-SVM	0.380	0.230	0.286	0.512	0.487	0.449	0.434	0.687
DirMix	0.441	**0.232**	0.303	**0.519**	0.501	0.470	0.443	0.697

Table 3.3: Experimental results for top 78 popular tags. CBA stands for Codeword Bernoulli Average (CBA) [51], HEM-GMM for hierarchical EM Gaussian Mixture Models [128], HEM-DTM for hierarchical EM Dynamic Texture Model [31].

Model	Precision	Recall	F-score	AROC	MAP	P10
CorrLog	0.48	**0.28**	**0.35**	**0.74**	**0.51**	**0.55**
CBA	0.41	0.24	0.29	0.69	0.47	0.49
HEM-GMM	**0.49**	0.23	0.26	0.66	0.45	0.47
HEM-DTM	0.47	0.25	0.30	0.69	0.48	0.53

The music data comes from CAL-500 Dataset [127]. There are 500 Western polyphonic songs and the annotations were collected from more than three human subjects per song. When training the classifier, we only use the binary annotations with $\{-1, 1\}$ to indicate whether the tag is relevant to the song. We are more interested in predicting more "useful" tags rather than very obscure ones. Following the same setting in [31, 92], we only evaluate on the 78 tags that have at least 50 songs and 97 top popular tags.

For song representation, we use the delta Mel-Frequency Cepstral Coefficient (MFCC) feature and the "bag-of-words" model [51]. First, we exact MFCC features for each song with a 23ms time window, and the 39-dimensional delta MFCC features are concatenated from the MFCC features, and their first and second derivatives. Normalization into zero mean and unit variance in each dimension is also applied to the MFCC features. After that, we utilize k-means to learn K cluster centroids as “audio dictionary”, and further obtain the “bag-of-words” representation of songs. The dictionary size K is set to 2000.

3.6.1 Experimental Setting

For fair comparison, we used the same experimental setting as in [31, 92], that is, we used 5-fold cross validation for performance evaluation, where in each round, we first learned the model parameters of CorrLog with the 400-song training set and then predicted confidence ratings on the remaining 100-song test set. The conditional probability, i.e., the confidence rating, of a tag being assigned to a song was obtained by calculating the marginal probability $p(\mathbf{y}_i|\mathbf{x}; \widetilde{\Theta})$ of the joint probability $p(\mathbf{y}|\mathbf{x}; \widetilde{\Theta})$ (3.4), with LBP. To compensate for non-uniform label prior, we adopted the same heuristic used in [51] by introducing a “diversity factor” $d = 1.25$. For each predicted confidence rating, we subtracted d times the mean confidence for that tag. We then assigned each song with the top 10 most confident tags.

Annotation was evaluated by mean precision and recall over the tags. Given the 10 annotations per song in the test set, we calculated precision and recall for each tag and then averaged across all considered tags. The final result was averaged over 5 rounds of cross validation. In addition, F-score, the harmonic mean of precision and recall, was computed to summarize the two aspects of precision and recall.

For retrieval, we first ranked the songs in the descending order according to confidence ratings for a specific tag. Better retrieval result corresponds to cases that more relevant songs appear at the top of the ranking list. Then, we calculated precision at every position down the ranking list via dividing the number of true positives found so far by the total number of songs so far. Evaluation was conducted through averaged precision and *precision at k* ($k = 3, 5, 10$) as in [92]. Averaged precision was computed by taking the average of all the positions down the ranking list where new true positives were found. Precision at k was k-th precision that we calculated on the ranking list.

3.6.2 Results and Dicussions

We compare our results with the state-of-the-art performance on the CAL-500 dataset. For the 97 tags setting, we compare with CBA [51], GMM [128], context-SVM [97] and Dirichlet mixtures (DirMix) [92]. Their results were originally reported in [92] and cited here in Table 3.2 for more convenient comparison. For the 78 tags setting, CBA, hierarchical EM Gaussian mixture models (HEM-GMM) and hierarchical EM dynamic texture model (HEM-DTM) [31] were compared. Their original results reported in [31] and copied in Table 3.3. The results of CorrLog with the elastic net regularization are also reported in Table 3.2 and Table 3.3, for the two settings respectively.

From Table 3.2, we can see that Context-SVM and DirMix generally outperform CBA and GMM. We believe this is due to the fact that the former two are able to utilize the information of label correlations. CorrLog futher improves the performance on most evaluation metrics. Similar results can be observed from Table 3.3 for the 78 tags setting.

3.7 Experiments: Multi-Label Image Classification

In this section, we apply the proposed CorrLog to multi-label image classification. In particular, four multi-label image datasets are used in the follow-

Table 3.4: Datasets summary. #images stands for the number of all images, #features stands for the dimension of the features, and #labels stands for the number of labels.

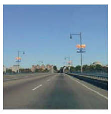
Datasets	#images	#features	#labels
MULANscene	2047	294	6
MITscene-PHOW	2688	3600	8
MITscene-CNN	2688	4096	8
PASCAL07-PHOW	9963	3600	20
PASCAL07-CNN	9963	4096	20
PASCAL12-PHOW	11540	3600	20
PASCAL12-CNN	11540	4096	20

ing experiments, including MULAN scene (MULANscene)[1], MIT outdoor scene (MITscene) [99], PASCAL VOC 2007 (PASCAL07) [41] and PASCAL VOC 2012 (PASCAL12) [40]. MULAN scene dataset contains 2047 images with 6 labels, and each image is represented by 294 features. MIT outdoor scene dataset contains 2688 images in 8 categories. To make it suitable for multi-label experiment, we transformed each category label with several tags according to the image contents of that category[2]. PASCAL VOC 2007 dataset consists of 9963 images with 20 labels. For PASCAL VOC 2012, we use the available train-validation subset which contains 11540 images. In addition, two kinds of features are adopted to represent the last three datasets, i.e., the PHOW (a variant of dense SIFT descriptors extracted at multiple scales) features [13] and deep CNN (convolutional neural network) features [23, 60]. Summary of the basic information of the datasets is illustrated in Table 4.1. To extract PHOW features, we use the VLFeat implementation [129]. For deep CNN features, we use the 'imagenet-vgg-f' model pretrained on ImageNet database [23] which is available in MatConvNet matlab toolbox [130].

[1] http://mulan.sourceforge.net/

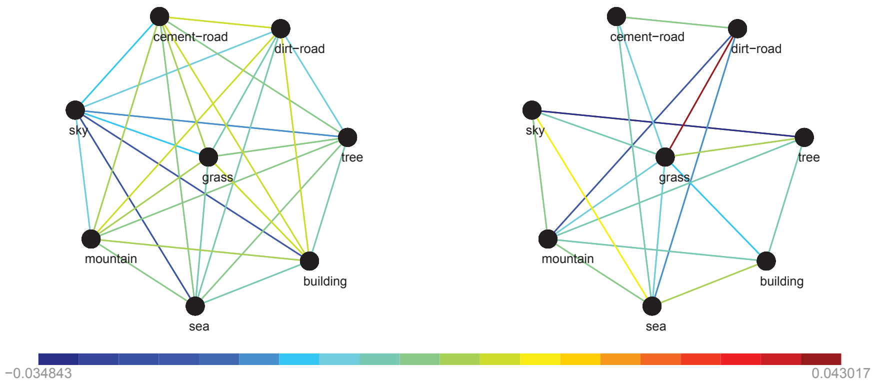
[2] The 8 categories are coast, forest, highway, insidecity, mountain, opencountry, street, and tallbuildings. The 8 binary tags are building, grass, cement-road, dirt-road, mountain, sea, sky, and tree. The transformation follows, $C1 \rightarrow (B6, B7)$, $C2 \rightarrow (B4, B8)$, $C3 \rightarrow (B3, B7)$, $C4 \rightarrow (B1)$, $C5 \rightarrow (B5, B7)$, $C6 \rightarrow (B2, B4, B7)$, $C7 \rightarrow (B1, B3, B7)$, $C8 \rightarrow (B1, B7)$. For example, coast ($C1$) is tagged with sea ($B6$) and sky ($B7$).

Table 3.5: Learned CorrLog label graph on MITscene using ℓ_2 or elastic net regularization.

MITscene Images and Tags							
coast 	forest 	highway 	inside-city 	mountain 	open-country 	street 	tall-building
sea sky	dirt-road tree	cement-road sky	building	mountain sky	grass dirt-road sky	building cement-road sky	building sky

Learned CorrLog Label Graph

ℓ_2 regularization | Elastic net regularization



3.7.1 A Warming-Up Qualitative Experiment

As an extension to ℓ_2 regularized CorrLog, the proposed method utilizes elastic net to inherit individual advantages of ℓ_2 and ℓ_1 regularization. To build up the intuition, we employ MITscene with PHOW features to visualize the difference between ℓ_2 and elastic net regularization. Table 3.5 presents the learned CorrLog label graphs using these two types of regularization respectively. In the label graph, the color of each edge represents the correlation strength between two certain labels. We have also listed 8 representative example images, one for each category, and their binary tags for completeness.

According to the comparison, one can see that elastic net regularization results in a sparse label graph due to its ℓ_1 component, while ℓ_2 regularization can only lead to a fully-connected label graph. In addition, the learned label correlations in elastic net case are more reasonable than that of ℓ_2. For example, in the ℓ_2 label graph, dirt-road and mountain have weekly positive correlation (according to the link between them), though they seldom co-occur on the images in the datasets, while in the elastic net graph, their correlation is corrected as negative. It has to be confessed that elastic net regularization also discarded some reasonable correlations such as cement-road and building. This phenomenon is a direct result of the compromise between learning stability and model sparsity. We shall mention that those reasonable correlations can be maintained by decreasing λ_1, λ_2 or ϵ, though more unreasonable connections will also be maintained. Thus, applying weak sparsity may impair the model performance. As a result, it is important to choose a good level of sparsity to achieve a compromise. In our experiments, CorrLog with elastic net regularization generally outperforms that with ℓ_2 regularization, which confirms our motivation that appropriate level of sparsity in feature selection and label correlations help boost the performance of multi-label classification. In the following presentation, we will use CorrLog with elastic net regularization in all experimental comparisons. To benefit following research, our code is available upon request.

3.7.2 Quantitative Experimental Setting

In this subsection, we present further comparisons between CorrLog and other multi-label classification methods. First, to demonstrate the effectiveness of utilizing label correlation, we first compare CorrLog's performance with ILRs. Moreover, four state-of-the-art multi-label classification methods - instance-based learning by logistic regression (IBLR) [28], multi-label k-nearest neighbour (MLKNN) [152], classifier chains (CC) [108] and maximum margin output coding (MMOC) [155] were also employed for comparison study. Note that ILRs can be regarded as the basic baseline and other methods represent state-of-the-arts. In our experiments, LIBlinear [42] ℓ_2-regularized logistic regression is employed to build

Table 3.6: MULANscene performance comparison via 5-fold cross validation. Marker $*/\circledast$ indicates whether CorrLog is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level).

Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
MULANscene	CorrLog	0.095±0.007	**0.341±0.020**	**0.710±0.018**	**0.728±0.017**	0.745±0.016	0.734±0.017
	ILRs	0.117±0.006 $*$	0.495±0.022 $*$	0.592±0.016 $*$	0.622±0.014 $*$	0.677±0.016 $*$	0.669±0.014 $*$
	IBLR	**0.085±0.004** $\circledast$	0.358±0.016	0.677±0.018 $*$	0.689±0.019 $*$	**0.747±0.010**	**0.738±0.014**
	MLKNN	0.086±0.003	0.374±0.015 $*$	0.668±0.018 $*$	0.682±0.019 $*$	0.742±0.013	0.734±0.012
	CC	0.104±0.005 $*$	0.346±0.015	0.696±0.015 $*$	0.710±0.015 $*$	0.716±0.018 $*$	0.706±0.014 $*$
	MMOC	0.126±0.017 $*$	0.401±0.046 $*$	0.629±0.049 $*$	0.639±0.050 $*$	0.680±0.031 $*$	0.638±0.049 $*$

binary classifiers for ILRs. As for other methods, we use publicly available codes in MEKA[1] or the authors' homepages.

We used six different measures to evaluate the performance. These include different loss functions (Hamming loss and zero-one loss) and other popular measures (accuracy, F1 score, Macro-F1 and Micro-F1). The details of these evaluation measures can be found in [29, 87, 108, 126]. The parameters for CorrLog are fixed across all experiments as $\lambda_1 = 0.001$, $\lambda_2 = 0.001$ and $\epsilon = 1$. On each dataset, all the methods are compared by 5-fold cross validation. The mean and standard deviation are reported for each criterion. In addition, paired t-tests at 0.05 significance level is applied to evaluate the statistical significance of performance difference.

3.7.3 Quantitative Results and Discussions

Tables 3.6, 3.7, 3.8 and 3.9 summarized the experimental results on MULANscene, MITscene, PASCAL07 and PASCAL12 of all six algorithms evaluated by the six measures. By comparing the results of CorrLog and ILRs, we can clearly see the improvements obtained by exploiting label correlations for multi-label classification. Except the Hamming loss, CorrLog greatly outperforms ILRs on all datasets. Especially, the reduction of zero-one loss is significant on all four

[1] http://meka.sourceforge.net/

Table 3.7: MITscene performance comparison via 5-fold cross validation. Marker ∗/⊛ indicates whether CorrLog is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level).

Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
MITscene-PHOW	CorrLog	0.045±0.006	**0.196±0.017**	**0.884±0.012**	**0.914±0.010**	0.883±0.017	**0.915±0.011**
	ILRs	0.071±0.002 ∗	0.358±0.015 ∗	0.825±0.007 ∗	0.877±0.005 ∗	0.833±0.007 ∗	0.872±0.003 ∗
	IBLR	0.060±0.003 ∗	0.243±0.021 ∗	0.845±0.012 ∗	0.879±0.008 ∗	0.848±0.009 ∗	0.886±0.006 ∗
	MLKNN	0.069±0.002 ∗	0.326±0.022 ∗	0.810±0.009 ∗	0.857±0.006 ∗	0.827±0.009 ∗	0.869±0.004 ∗
	CC	0.047±0.005	0.200±0.021	0.883±0.012	0.913±0.008	**0.883±0.015**	0.913±0.009
	MMOC	0.062±0.010 ∗	0.274±0.035 ∗	0.845±0.017 ∗	0.885±0.014 ∗	0.846±0.024 ∗	0.885±0.017 ∗
MITscene-CNN	CorrLog	**0.017±0.004**	0.088±0.015	0.953±0.008	0.966±0.006	**0.957±0.011**	**0.968±0.006**
	ILRs	0.020±0.002 ∗	0.102±0.015 ∗	0.947±0.006 ∗	0.962±0.004 ∗	0.951±0.007 ∗	0.963±0.005 ∗
	IBLR	0.022±0.001 ∗	0.090±0.009	0.944±0.004	0.957±0.003 ∗	0.944±0.004 ∗	0.958±0.003 ∗
	MLKNN	0.024±0.002 ∗	0.104±0.005 ∗	0.939±0.003 ∗	0.954±0.003 ∗	0.941±0.002 ∗	0.955±0.004 ∗
	CC	0.021±0.003 ∗	0.075±0.008 ⊛	0.951±0.005	0.962±0.004 ∗	0.948±0.007 ∗	0.961±0.005 ∗
	MMOC	0.018±0.002	**0.062±0.005** ⊛	**0.959±0.003** ⊛	**0.967±0.003**	0.955±0.005	0.967±0.004

datasets with different type of features. This confirms the value of correlation modelling to joint prediction. However, it should be noticed that the improvement of CorrLog over ILRs is less significant when the performance is measured by Hamming loss. This is because Hamming loss treats the prediction of each label individually.

In addition, CorrLog is more effective in exploiting label correlations than other four state-of-the-art multi-label classification algorithms. For MULAN-scene dataset, CorrLog achieved comparable results with IBLR and both of them outperformed other methods. For MITscene dataset, both PHOW and CNN features are very effective representations and boost the classification results. As a consequence, the performance of CorrLog and the four multi-label classification algorithms are very close to each other. It is worth noting that, the MMOC method is time-consuming in the training stage, though it achieved the best performance on this dataset. As for both PASCAL07 and PASCAL12 datasets, CNN features perform significantly better than PHOW features. CorrLog obtained much better results than the competing multi-label classification schemes, except for the Hamming loss and zero-one loss. Note that the CorrLog also performs competitively with PLEM and CGM, according to the results reported

Table 3.8: PASCAL07 performance comparison via 5-fold cross validation. Marker ∗/⊛ indicates whether CorrLog is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level).

Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
PASCAL07-PHOW	CorrLog	0.068±0.001	**0.776±0.007**	**0.370±0.010**	**0.423±0.012**	**0.367±0.011**	**0.480±0.008**
	ILRs	0.093±0.001 ∗	0.878±0.007 ∗	0.294±0.008 ∗	0.360±0.009 ∗	0.332±0.008 ∗	0.404±0.007 ∗
	IBLR	0.066±0.001 ⊛	0.832±0.003 ∗	0.270±0.005 ∗	0.308±0.006 ∗	0.258±0.007 ∗	0.408±0.009 ∗
	MLKNN	0.066±0.001 ⊛	0.839±0.006 ∗	0.256±0.007 ∗	0.291±0.008 ∗	0.235±0.006 ∗	0.392±0.007 ∗
	CC	0.091±0.000 ∗	0.845±0.010 ∗	0.318±0.005 ∗	0.379±0.003 ∗	0.348±0.004 ∗	0.417±0.001 ∗
	MMOC	**0.065±0.001** ⊛	0.850±0.003 ∗	0.259±0.009 ∗	0.299±0.011 ∗	0.206±0.007 ∗	0.392±0.012 ∗
PASCAL07-CNN	CorrLog	0.038±0.001	0.516±0.010	**0.642±0.010**	**0.696±0.010**	**0.674±0.002**	**0.724±0.006**
	ILRs	0.046±0.001 ∗	0.574±0.011 ∗	0.610±0.010 ∗	0.673±0.009 ∗	0.651±0.004 ∗	0.688±0.007 ∗
	IBLR	0.043±0.001 ∗	0.554±0.011 ∗	0.597±0.014 ∗	0.649±0.015 ∗	0.621±0.007 ∗	0.682±0.010 ∗
	MLKNN	0.043±0.001 ∗	0.557±0.010 ∗	0.585±0.014 ∗	0.635±0.015 ∗	0.613±0.006 ∗	0.668±0.011 ∗
	CC	0.051±0.001 ∗	0.586±0.008 ∗	0.602±0.008 ∗	0.668±0.008 ∗	0.635±0.009 ∗	0.669±0.008 ∗
	MMOC	**0.037±0.000** ⊛	**0.512±0.008**	0.634±0.009 ∗	0.684±0.009 ∗	0.663±0.005 ∗	0.719±0.004 ∗

Table 3.9: PASCAL12 performance comparison via 5-fold cross validation. Marker ∗/⊛ indicates whether CorrLog is statistically superior/inferior to the compared method (using paired t-test at 0.05 significance level).

Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
PASCAL12-PHOW	CorrLog	0.070±0.001	**0.790±0.009**	**0.344±0.009**	**0.393±0.010**	**0.369±0.014**	**0.449±0.006**
	ILRs	0.100±0.001 ∗	0.891±0.009 ∗	0.269±0.007 ∗	0.333±0.008 ∗	0.324±0.008 ∗	0.370±0.005 ∗
	IBLR	0.068±0.001 ⊛	0.869±0.009 ∗	0.219±0.005 ∗	0.252±0.003 ∗	0.253±0.007 ∗	0.345±0.005 ∗
	MLKNN	0.069±0.001 ⊛	0.883±0.008 ∗	0.191±0.006 ∗	0.218±0.005 ∗	0.213±0.007 ∗	0.306±0.006 ∗
	CC	0.097±0.001 ∗	0.862±0.012 ∗	0.291±0.010 ∗	0.350±0.010 ∗	0.340±0.007 ∗	0.380±0.006 ∗
	MMOC	**0.067±0.001** ⊛	0.865±0.003 ∗	0.227±0.005 ∗	0.262±0.007 ∗	0.200±0.007 ∗	0.346±0.004 ∗
PASCAL12-CNN	CorrLog	0.040±0.001	0.526±0.010	**0.639±0.007**	**0.695±0.007**	**0.674±0.006**	**0.708±0.006**
	ILRs	0.051±0.001 ∗	0.613±0.002 ∗	0.581±0.005 ∗	0.649±0.006 ∗	0.638±0.005 ∗	0.658±0.005 ∗
	IBLR	0.045±0.001 ∗	0.574±0.006 ∗	0.575±0.009 ∗	0.627±0.010 ∗	0.613±0.008 ∗	0.657±0.006 ∗
	MLKNN	0.045±0.002 ∗	0.575±0.012 ∗	0.566±0.015 ∗	0.616±0.017 ∗	0.604±0.011 ∗	0.645±0.013 ∗
	CC	0.055±0.001 ∗	0.615±0.010 ∗	0.579±0.009 ∗	0.647±0.010 ∗	0.623±0.005 ∗	0.643±0.007 ∗
	MMOC	**0.039±0.001** ⊛	**0.525±0.005**	0.619±0.006 ∗	0.669±0.007 ∗	0.659±0.004 ∗	0.699±0.005 ∗

in [121].

3.7.4 Complexity Analysis and Execution Time

Table 3.10 summarizes the algorithm computational complexity of all multi-label classification methods. The training computational cost of both CorrLog and ILRs are linear to the number of labels, while CorrLog causes more testing computational cost than ILRs due to the iterative belief propagation algorithm. In contrast, the training complexity of CC and MMOC are polynomial to the number of labels. The two instance-based methods, MLKNN and IBLR, are relatively computational in both train and test stages due to the involvement of instance-based searching of nearest neighbours. In particular, training MLKNN requires estimating the prior label distribution from training data which needs the consideration of all k nearest neighbours of all training samples. Testing a given sample in MLKNN consists of finding its k-nearest neighbours and applying maximum a posterior (MAP) inference. Different from MLKNN, IBLR constructs logistic regression models by adopting labels of k-nearest neighbours as features.

To evaluate the practical efficiency, Tables 3.11 and 3.12 present the execution time (train and test phase) of all comparison algorithms under Matlab environment. A Linux server equipped with Intel Xeon CPU (8 cores @ 3.4 GHz) and 32 GB memory is used for conducting all the experiments. CorrLog is implemented in Matlab language, while ILRs is implemented based on LIBlinear's mex functions. MMOC is evaluated using the authors' Matlab code which also builds upon LIBlinear. As for IBLR, MLKNN and CC, the MEKA Java library is called via a Matlab wrapper. Based on the comparison results, the following observations can be made: 1) the execution time is largely consistent with the complexity analysis, though there maybe some unavoidable computational differences between Matlab scripts, mex functions and Java codes; 2) CorrLog's train phase is very efficient and its test phase is also comparable with ILRs, CC and MMOC; 3) CorrLog is more efficient than IBLR and MLKNN in both train and test stages.

3.8 Summary

We have proposed a new multi-label classification algorithm CorrLog and applied it to multi-label image classification. Built upon IRLs, CorrLog explicitly models the pairwise correlation between labels, and thus improves the effectiveness for multi-label classification. Besides, by using the elastic net regularization, CorrLog is able to exploit the sparsity in both feature selection and label correlations, and

Table 3.10: Computational complexity analysis. Recall that n stands for the number of train images, D stands for the dimension of the features, and m stands for the number of labels. Note that C is the iteration number of the max-product algorithm in CorrLog, and K is the number of nearest neighbours in MLKNN and IBLR.

Methods	Train	Test per image
CorrLog	$\mathcal{O}(nDm)$	$\mathcal{O}(Dm + Cm^2)$
ILRs	$\mathcal{O}(nDm)$	$\mathcal{O}(Dm)$
IBLR	$\mathcal{O}(Kn^2Dm + nDm)$	$\mathcal{O}(KnDm + Dm)$
MLKNN	$\mathcal{O}(Kn^2Dm)$	$\mathcal{O}(KnDm)$
CC	$\mathcal{O}(nDm + nm^2)$	$\mathcal{O}(Dm + m^2)$
MMOC	$\mathcal{O}(nm^3 + nDm^2 + n^4)$	$\mathcal{O}(m^3)$

Table 3.11: Average execution time (in seconds) comparison on MULANscene and MITscene.

	MULANscene		MITscene-PHOW		MITscene-CNN	
	Train	Test	Train	Test	Train	Test
CorrLog	0.09	1.74	2.80	2.12	2.46	2.08
ILRs	2.54	0.02	39.50	0.37	7.50	0.15
IBLR	12.01	2.63	218.98	53.31	215.28	52.19
MLKNN	10.29	2.36	188.08	45.87	176.52	42.78
CC	5.48	0.06	40.71	0.55	26.64	0.65
MMOC	851.98	0.51	2952.77	0.70	2162.13	0.48

Table 3.12: Average execution time (in seconds) comparison on PASCAL07 and PASCAL12.

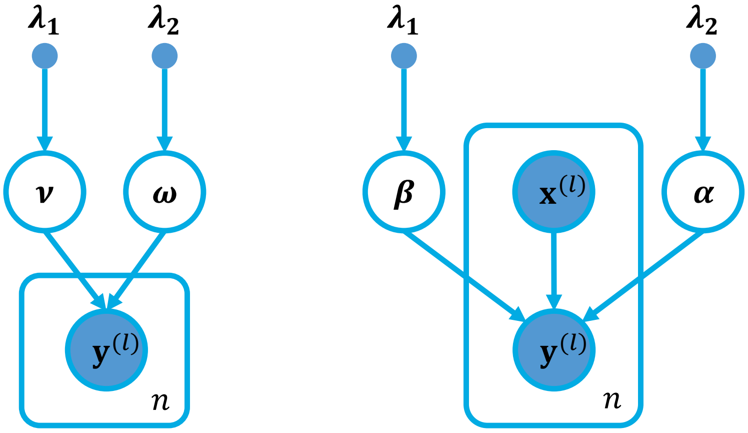
	PASCAL07-PHOW		PASCAL07-CNN		PASCAL12-PHOW		PASCAL12-CNN	
	Train	Test	Train	Test	Train	Test	Train	Test
CorrLog	8.94	10.68	8.35	11.08	9.67	12.58	8.62	13.06
ILRs	872.77	4.79	122.73	1.56	1183.45	5.59	161.71	1.83
IBLR	3132.18	779.15	2833.94	688.53	4142.75	1034.86	3824.06	947.90
MLKNN	2507.51	628.61	2232.19	551.26	3442.21	863.17	3020.29	779.50
CC	74315.65	7.48	8746.82	8.15	137818.99	8.38	15926.62	9.57
MMOC	86714.47	33.08	38403.54	17.75	97856.16	31.43	45541.01	20.66

thus further boost the performance of multi-label classification. Theoretically, we have shown that the generalization error of CorrLog is upper bounded and is independent of the number of labels. This suggests the generalization bound holds with high confidence even when the number of labels is large. Evaluations on benchmark music annotation and multi-label image datasets confirm the effectiveness of CorrLog for multi-label classification and show its competitiveness with the state-of-the-arts.

Chapter 4

Conditional Graphical Lasso for Multi-Label Classification

Multi-label image classification aims to predict multiple labels for a single image which contains diverse content. By utilizing label correlations, various techniques have been developed to improve classification performance. However, current existing methods either neglect image features when exploiting label correlations or lack the ability to learn image-dependent conditional label structures. In this chapter, we develop conditional graphical lasso (CGL) to handle these challenges. CGL provides a unified Bayesian framework for structure and parameter learning conditioned on image features. We formulate the multi-label prediction as CGL inference problem, which is solved by a mean field variational approach. Meanwhile, CGL learning is efficient due to a tailored proximal gradient procedure by applying the maximum a posterior (MAP) methodology. CGL performs competitively for multi-label image classification on benchmark datasets MULAN scene, PASCAL VOC 2007 and PASCAL VOC 2012, compared with the state-of-the-art multi-label classification algorithms.



(a) Graphical Lasso (b) Conditional Graphical Lasso

Figure 4.1: Comparison of graphical models between unconditional and conditional graphical Lasso. The templates denotes replica of n training images and labels. $\mathbf{x}^{(l)}$ represents the l-th image and $\mathbf{y}^{(l)}$ denotes its label vector. The parameters $\{\boldsymbol{\nu}, \boldsymbol{\omega}\}$, $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ are shared across training data, and are themselves parameterized by hyperparameters λ_1 and λ_2. In graphical Lasso, $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ parameterize unary and pairwise potentials, respectively. In contrast, the parameterization is achieved by considering linear functions of $\mathbf{x}^{(l)}$, i.e., $\boldsymbol{\beta}^T\mathbf{x}^{(l)}$ and $\boldsymbol{\alpha}^T\mathbf{x}^{(l)}$, in conditional graphical Lasso.

4.1 Introduction

Multi-label image classification targets the specific problem of predicting the presence or absence of multiple object categories in an image. Like other high-level vision tasks such as object recognition [5], image annotation [43] and scene classification [15], multi-label image classification is very challenging due to large intra-class variation caused by viewpoint, scale, occlusion, illumination, etc. To meet these challenges, many image representation and feature learning schemes have been developed to gain variation-invariance, such as GIST [99], dense SIFT [13], VLAD [55], object bank [72], and deep CNN [23, 60]. Meanwhile, label correlations, which are typically encoded in a graph structure, have been exploited to further improve classification performance.

In literature, the task of finding a meaningful label structure is commonly handled with probabilistic graphical models [58]. A classical approach is the ChowLiu

Tree [30] which utilizes mutual information between labels to obtain a maximum spanning tree structure and is proved to be equivalent to the maximum likelihood estimation. Recently, probabilistic label enhancement model (PLEM) [77] exploits label co-occurrence pairs based on a maximum spanning tree construction and applies the tree structure to solve multi-label classification problem. In these methods, the structure learned on labels is naively used to model the label structure conditioned on features, which is inappropriate because this kind of structure describes the label distribution rather than the conditional distribution of labels.

To target the problem, several methods have been proposed to incorporate input features during label structure learning [16, 121, 151]. An extension to the ChowLiu Tree is designed in [16] which investigates two kinds of conditional mutual information to learn a conditional tree structure. Meanwhile, a conditional directed acyclic graph (DAG) is also designed to reformulate multi-label classification into a series of single-label classification problems [151]. More recently, clique generating machine (CGM) [121] learns the conditional label structure in a structured support vector machine framework. These methods assume a shared label graph across all input images, which provides a better approximation to the true structure than the unconditional label graph. However, such a shared conditional graph is not flexible enough to characterize the label structure of each unique image.

In this study, we propose a conditional label structure learning method which can produce image-dependent conditional label structures. Our method extends the classical graphical lasso (GL) framework which estimates graph structure associated with Markov random field (MRF) by employing sparse constraints [68, 91, 105]. [1] We term the proposed method as conditional graphical lasso (CGL).

[1] In literature, the term "graphical Lasso" is traditionally restricted to refer structure learning for (continuous) Gaussian MRF only. In this chapter, we use this concept to cover continuous, discrete and mixed random fields.

See Figure 4.1 for the comparison between graphical models of GL and CGL. CGL offers a principled approach to model conditional label structures within a unified Bayesian framework. Besides, CGL provides a simple but effective way to learn image-dependent label structures by considering conditional label correlations as linear weight functions of features. Such favourable properties are achieved via an efficient mean field approximate inference procedure and a tailored proximal gradient based learning algorithm.

The rest of this chapter is organized as follows. Section 4.2 briefly reviews related multi-label classification methods from the viewpoint of learning strategy. Section 4.3 first introduces the GL framework for modeling binary valued labels and then describes the proposed conditional extension CGL in terms of a probabilistic perspective. Section 4.4 presents algorithms for CGL inference and learning, of which inference is achieved by a mean-field variational approach and learning is achieved by a proximal gradient method. Section 4.5 reports results of empirical evaluations, where both qualitative and quantitative experiments are considered on several benchmark multi-label image classification datasets.

4.2 Related Works

Apart from the structure learning approach, we briefly review three other main categories of multi-label classification methods which follows the taxonomy of recent surveys [50,125,153]. The three categories include problem transformation, algorithm adaptation and dimension reduction.

Problem transformation methods reformulate multi-label classification into single-label classification. For example, the binary relevance (BR) method trains binary classifiers for each label independently. By considering label dependency, classifier chain (CC) [108], as well as its ensemble and probabilistic variants [29], constructs a chain of binary classifiers, in which each classifier additionally use the previous labels as its input features. Another group of algorithms are built upon label powerset or hierarchy information, which includes random k-label sets (RAKEL) [126], pruned problem transformation (PPT) [107], hierarchical binary relevance (HBR) [22] and hierarchy of multi-label classifiers (HOMER) [125].

Algorithm adaptation methods extend typical classifiers to multi-label situation. For example, multi-label k-nearest neighbour (MLKNN) [152] adapts KNN to handle multi-label classification, which exploits the prior label distribution within the neighbourhood of an image instance and applies the maximum a posterior (MAP) prediction. Instance based logistic regression (IBLR) [28] adapts LR by utilizing label information from the neighbourhood of an image instance as features.

Dimension reduction methods target to handle high-dimensional features and labels. The reduction of feature space aims to reduce feature dimension either by feature selection or by feature extraction. For example, multi-label informed latent semantic indexing (MLSI) [146], multi-label least square (MLLS) [56], multi-label F-statistics (MLF) and multi-label ReliefF (MLRF) [59]. Label specific features (LIFT) [150] method represents an image instance as its distances to label-specific clustering centers of positive and negative training image instances, and use the features to train binary classifiers and make predictions. On the other hand, the reduction of label space utilizes a variety of strategies, such as multi-label compressed sensing (MLCS) [53], compressed labeling (CL) [156], principal label space transformation (PLST) [120] and its conditional variant [26], canonical correlation analysis output coding (CCA-OC) [154], and maximum margin output coding (MMOC) [155].

4.3 Model Representation

In this section, we first review the basic GL framework from a Bayesian perspective. Then we present the extension by considering conditional variables and exploiting a group sparse prior. To simplify discussion, we will consider a fully-connected and pairwise label graph, though the same methodology can be easily applied to a higher-order case.

4.3.1 Graphical Lasso

An GL framework considers the problem of estimating the graph structure associated with an MRF. Consider the ℓ_1-regularized Ising MRF [105] over a label vector $\mathbf{y} \in \{-1, 1\}^m$, GL employs an ℓ_1 regularization over pairwise parameters and achieves conditional independence by increasing sparsity. An ℓ_1 regularization is equivalent to imposing a Laplacian prior. Thus, we can formulate the ℓ_1-regularized Ising model into the Bayesian framework which is given by

$$p(\mathbf{y}, \boldsymbol{\nu}, \boldsymbol{\omega}) = p(\mathbf{y}|\boldsymbol{\nu}, \boldsymbol{\omega})p(\boldsymbol{\nu})p(\boldsymbol{\omega}), \tag{4.1}$$

$$p(\mathbf{y}|\boldsymbol{\nu}, \boldsymbol{\omega}) \propto \exp\left\{\sum_{i=1}^{m} \nu_i \mathbf{y}_i + \sum_{i<j} \omega_{ij}\mathbf{y}_i\mathbf{y}_j\right\}, \tag{4.2}$$

$$p(\boldsymbol{\nu}) \propto \lambda_1^{d/2} \exp(-\lambda_1 \|\boldsymbol{\nu}\|_2^2), \tag{4.3}$$

$$p(\boldsymbol{\omega}) \propto \lambda_2^{d/2} \exp(-\lambda_2 \|\boldsymbol{\omega}\|_1), \tag{4.4}$$

where $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ parameterize the unary and pairwise potentials over $\mathbf{y}$. λ_1 and λ_2 are hyperparameters which control the strength of regularization over $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$, respectively. Though the label graph learned by GL can be applied to multi-label classification, both $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ have no explicit connection to the image features. In the next subsection, we will make a conditional extension to GL by incorporating image features to the learning process of label graph which leads to our CGL framework.

4.3.2 Conditional Graphical Lasso

As an extension to the GL framework, we consider a more deliberate structure learning approach when conditional variables emerge. In particular, CGL framework aims to search adaptive structures among response variables (labels) conditioned on input variables (image features).

For the particular multi-label classification task, we study the problem of learning a joint prediction $\mathbf{y} = f_\Theta(\mathbf{x}) : \mathcal{X} \mapsto \mathcal{Y}$, where the prediction function f is parameterized by Θ, the image feature space $\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\| \leq 1, \mathbf{x} \in \mathbb{R}^d\}$ and the label space $\mathcal{Y} = \{-1, 1\}^m$. By considering appropriate priors on Θ, we arrive at the joint probability distribution over $\mathbf{y}$ and Θ conditioned on $\mathbf{x}$,

$$p(\mathbf{y}, \Theta|\mathbf{x}) = p(\mathbf{y}|\mathbf{x}, \Theta)p(\Theta). \tag{4.5}$$

Note that the joint conditional distribution can be specified according to certain considerations, such as dealing with overfitting problems and inducing sparsity over label correlations.

Consider a label graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, 2, \cdots, m\}$ denotes the set of nodes corresponding to labels and $\mathcal{E} = \{(i, j) : i < j; i, j \in \mathcal{V}\}$ represents the set of edges encoding pairwise label correlations. We can model the conditional distribution $p(\mathbf{y}|\mathbf{x}, \Theta)$ with a set of unary and pairwise potentials over the label graph $\mathcal{G}$,

$$p(\mathbf{y}|\mathbf{x}, \Theta) \propto \exp\left\{\sum_{i=1}^{m} \nu_i(\mathbf{x})\mathbf{y}_i + \sum_{i<j} \omega_{ij}(\mathbf{x})\mathbf{y}_i\mathbf{y}_j\right\}. \tag{4.6}$$

The above unary and pairwise weights $\{\nu_i(\mathbf{x})\}$, $\{\omega_{ij}(\mathbf{x})\}$ can be linear or nonlinear functions of $\mathbf{x}$. For simplicity, we restrict the weights to be linear functions of $\mathbf{x}$ which are defined as

$$\begin{cases} \nu_i(\mathbf{x}) = \beta_i^T \mathbf{x}, & \text{for } i \in \mathcal{V}; \\ \omega_{ij}(\mathbf{x}) = \alpha_{ij}^T \mathbf{x}, & \text{for } (i, j) \in \mathcal{E}. \end{cases} \tag{4.7}$$

To this end, the model parameter $\Theta = \{\boldsymbol{\beta}, \boldsymbol{\alpha}\}$ contains $\boldsymbol{\beta} = [\beta_1, ..., \beta_m]$ and $\boldsymbol{\alpha} = [\alpha_{12}, ..., \alpha_{(m-1)m}]$. Note that, conditioned on $\mathbf{x}$, (4.6) is exactly an Ising model for $\mathbf{y}$. It can also be treated as a special instantiation of CRF [63], by defining features $\phi_i(\mathbf{x}, \mathbf{y}) = \mathbf{y}_i\mathbf{x}$ and $\psi_{ij}(\mathbf{x}, \mathbf{y}) = \mathbf{y}_i\mathbf{y}_j\mathbf{x}$.

As for the model prior $p(\Theta)$, we employ multivariate d-dimensional Gaussian priors over each group of the node regression coefficients, which is equivalent to place an ℓ_2-norm regularizer on the nodewise parameters $\boldsymbol{\beta}$. Meanwhile, we use multivariate d-dimensional Multi-Laplacian priors [104] over each group of the edge regression coefficients, which can be regarded as imposing an $\ell_{2,1}$-norm, i.e., group-Lasso regularizer on the edgewise parameters $\boldsymbol{\alpha}$. More specifically,

$$p(\Theta) = p(\boldsymbol{\beta})p(\boldsymbol{\alpha}) = \prod_{i=1}^{m} p(\beta_i) \prod_{i<j} p(\alpha_{ij}), \tag{4.8}$$

$$p(\beta_i) \propto \lambda_1^{d/2} \exp(-\lambda_1 \|\beta_i\|_2^2), \tag{4.9}$$

$$p(\alpha_{ij}) \propto \lambda_2^{d/2} \exp(-\lambda_2 \|\alpha_{ij}\|_2), \tag{4.10}$$

where hyperparameters λ_1 and λ_2 control the strength of regularization over $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, respectively. It is worth mentioning that one can also choose other kinds of priors over the model parameters provided the priors can induce certain sparsity over pairwise correlations.

It is interesting to compare the differences between GL and CGL in modeling the label correlations. Firstly, GL uses scalar parameters to define the pattern and strength while CGL uses parametric functions of input features. Secondly, though both GL and CGL produce a shared label structure pattern for different instances, CGL additionally has the flexibility of capturing the varying correlation strength for different instances.

4.4 Algorithms

In this section, we derive both inference and learning algorithms for CGL. Generally, the label space $\mathcal{Y} = \{-1, 1\}^m$ in (4.6) maintains an exponentially large number of possible configurations. To normalize the conditional distribution in

(4.6), one requires the log-partition function. For CGL with linear weight functions of $\mathbf{x}$ in (4.7), the log-partition function is defined as

$$A(\Theta, \mathbf{x}) = \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp \left\{ \sum_{i=1}^{m} \mathbf{y}_i \beta_i^T \mathbf{x} + \sum_{i<j} \mathbf{y}_i \mathbf{y}_j \alpha_{ij}^T \mathbf{x} \right\}, \tag{4.11}$$

which involves a summation over all the configurations. Hence, it is computationally intractable to exactly calculate the log-partition function. To make CGL inference and learning tractable, we resort to approximate inference and learning algorithms via the variational methodology.

4.4.1 Approximate Inference

Inference of CGL involves two main tasks: marginal inference and the most probable explanation (MPE) prediction. However, conducting inference from the exact distribution $p(\mathbf{y}|\mathbf{x})$ is intractable due to the log-partition function $A(\Theta, \mathbf{x})$. Considering tractable approximation techniques, we choose the variational approach instead of sampling methods for its simplicity and efficiency. In particular, by applying the mean field assumption, the optimal variational approximation of $p(\mathbf{y}|\mathbf{x})$ is obtained by

$$\widehat{q}(\mathbf{y}) = \arg \min_{\substack{q(\mathbf{y})= \\ \prod_i q(\mathbf{y}_i)}} \mathrm{KL}[q(\mathbf{y}) \| p(\mathbf{y}|\mathbf{x}, \Theta)]. \tag{4.12}$$

According to [12], the marginal $q(\mathbf{y}_i)$ that minimizes (4.12) is achieved by analytically minimizing a Lagrangian which consists of the Kullback-Leibler divergence and Lagrangian multipliers constraining the marginal $q(\mathbf{y}_i)$ to be a valid probability distribution. For brevity of presentation, we simply give the update formula for each $q(\mathbf{y}_i)$,

$$q(\mathbf{y}_i) \leftarrow \frac{1}{Z_i} \exp \mathbb{E}_{q(\mathbf{y}_{\setminus i})}[\ln p(\mathbf{y}|\mathbf{x}, \Theta)], \tag{4.13}$$

Algorithm 2 CGL Inference

```
Input: Image $\mathbf{x}$ and model parameters $\Theta = (\boldsymbol{\beta}, \boldsymbol{\alpha})$.
Output: Variational distribution $\widehat{q}(\mathbf{y}) = \prod_i \widehat{q}(\mathbf{y}_i)$.
Initialize $q^{(0)}(\mathbf{y}_i) \leftarrow \frac{1}{1+\exp\left\{-2\mathbf{y}_i\beta_i^T\mathbf{x}\right\}}$ for each $i$.
while not converged do
  for $i = 1, \cdots, m$ do
    Prepare expected statistics,
        $\xi_q(\mathbf{y}_{\setminus i}) = \left\{ \begin{array}{c} \mathbb{E}_{q^{(t+1)}(\mathbf{y}_j)}[\mathbf{y}_j] : 1 \leq j < i; \\ \mathbb{E}_{q^{(t)}(\mathbf{y}_j)}[\mathbf{y}_j] : i < j \leq m. \end{array} \right\}$
    Update the variational distribution $q^{(t+1)}(\mathbf{y}_i)$ with $\xi_q(\mathbf{y}_{\setminus i})$ by using (4.16).
    Update the $i$-th expected statistic $\mathbb{E}_{q^{(t+1)}(\mathbf{y}_i)}[\mathbf{y}_i]$.
  end for
  $t = t + 1$
end while
```

where $\mathbb{E}_p[g]$ calculates the expectation of function g w.r.t. distribution p, Z_i is the normalization term for distribution $q(\mathbf{y}_i)$, and we defined $q(\mathbf{y}_{\setminus i}) = \prod_{j \neq i} q(\mathbf{y}_j)$.

To solve (4.12) for updating $q(\mathbf{y}_i)$, we expand and reformulate the expectation w.r.t. $q(\mathbf{y}_{\setminus i})$. By dissecting out all the terms that contain $\mathbf{y}_i$, we obtain

$$\begin{aligned} & \mathbb{E}_{q(\mathbf{y}_{\setminus i})}[\ln p(\mathbf{y}|\mathbf{x}, \Theta)] \\ & = \mathbf{y}_i\beta_i^T\mathbf{x} + \mathbf{y}_i\mathbb{E}_{q(\mathbf{y}_{\setminus i})}\left[\sum_{j \neq i}\mathbf{y}_j\right]\alpha_{ij}^T\mathbf{x} + \text{const} && (4.14) \\ & = \mathbf{y}_i\beta_i^T\mathbf{x} + \mathbf{y}_i\sum_{j \neq i}\mathbb{E}_{q(\mathbf{y}_j)}[\mathbf{y}_j]\alpha_{ij}^T\mathbf{x} + \text{const}, && (4.15) \end{aligned}$$

where we have applied the marginalization property of the joint distribution $q(\mathbf{y}_{\setminus i})$ to obtain (4.15).

With a further consideration for the normalization constraint of a valid probability distribution, we arrive at a logistic regression for each $q(\mathbf{y}_i)$ given by

$$q(\mathbf{y}_i) = \sigma\left(2\mathbf{y}_i\left(\beta_i^T\mathbf{x} + \sum_{j \neq i}\mathbb{E}_{q(\mathbf{y}_j)}[\mathbf{y}_j]\alpha_{ij}^T\mathbf{x}\right)\right), \tag{4.16}$$

Algorithm 3 CGL Learning

```
Input: Training images and labels $\{\mathbf{X}, \mathbf{Y}\}$, hyperparameters $\{\lambda_1, \lambda_2\}$, and learning rate $\eta$, where $1/\eta$ is set larger than the Lipschitz constant of $\nabla J_s(\Theta)$ (4.25).
Output: Model parameters $\widehat{\Theta} = (\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\alpha}})$.
Initialize $\boldsymbol{\beta}^{(0)} = \mathbf{0}$, $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$.
while not converged do
  Update the variational distributions $\{\widehat{q}(\mathbf{y}^{(l)})\}_{l=1}^n$ with $\Theta^{(k)} = (\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)})$ by using Algorithm 2.
  Calculate the gradient of $J_s(\Theta)$ at $\Theta^{(k)} = (\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)})$ according to (4.24).
  Update $\Theta^{(k+1)} = (\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})$ by using (4.27);
    $k = k + 1$
end while
```

where $\sigma(t) = \frac{1}{1+\exp(-t)}$ is the sigmoid function. This formula requires the expectation of other variables connected to variable $\mathbf{y}_i$. Thus, a cycling and iterative updating for each $q(\mathbf{y}_i)$ is performed until convergence to a stationary point. Algorithm 2 presents the pseudo code for this procedure. It is worth mentioning that, we employed the most recent expected statistics $\xi_q(\mathbf{y}_{\setminus i})$ instead of the terms from previous round when updating one particular factor distribution $q(\mathbf{y}_i)$. This strategy can avoid undesired abrupt oscillations of the iterative procedure to some extend.

So far, it seems that our derivation only considers optimizing a factorized variational distribution $q(\mathbf{y})$ which approximates $p(\mathbf{y}|\mathbf{x})$. However, the same methodology can be straightforwardly applied to other inference and learning tasks. Take MPE for example, suppose we are given a new image $\mathbf{x}$, MPE aims to perform a joint prediction of its label vector $\mathbf{y}$ with some learned model parameter $\widehat{\Theta}$. Instead of conducting the max-product algorithm over $p(\mathbf{y}|\mathbf{x}, \widehat{\Theta})$, we can achieve the prediction $\widehat{\mathbf{y}}$ directly from $q(\mathbf{y})$.

4.4.2 Structure and Parameter Learning

Given a set of i.i.d. training images $\mathbf{X} = \{\mathbf{x}^{(l)}\}_{l=1}^n$ and their label vectors

$\mathbf{Y} = \{\mathbf{y}^{(l)}\}_{l=1}^{n}$, structure and parameter learning of CGL aims to find the optimal model parameter $\widehat{\Theta}$ which achieves the maximum a posterior (MAP) under certain values of hyperparameters $\{\lambda_1, \lambda_2\}$. It is worth emphasizing that the graphical structure is implicitly represented by the ℓ_2-norm of α_{ij}. In other words, a nonzero vector α_{ij} almost probably indicates an edge in the graph between node i and j, while a zero vector α_{ij} implies no such edge. To utilize the MAP methodology for CGL learning, the Bayesian rule is applied to obtain

$$\begin{aligned}\widehat{\Theta} &= \arg\max_{\Theta} p(\Theta|\mathbf{Y}, \mathbf{X}) && (4.17)\\ &= \arg\max_{\Theta} \frac{p(\mathbf{Y}, \Theta|\mathbf{X})}{\int_{\Theta} p(\mathbf{Y}, \Theta|\mathbf{X})} && (4.18)\\ &= \arg\max_{\Theta} p(\mathbf{Y}, \Theta|\mathbf{X}) && (4.19)\\ &= \arg\max_{\Theta} \prod_{l=1}^{n} p(\mathbf{y}^{(l)}|\mathbf{x}^{(l)}, \Theta)p(\Theta). && (4.20)\end{aligned}$$

Note that we have exploited the fact that the evidence $\int_{\Theta} p(\mathbf{Y}, \Theta|\mathbf{X})$ is independent of the model parameter Θ. And the final optimization problem (4.20) is achieved by considering (4.5) and the i.i.d. assumption.

By taking negative logarithm of the posterior and substituting (4.6), (4.9) and (4.10) into (4.20), the original maximization problem can be reformulated into an equivalent minimization problem as below,

$$\begin{aligned}\widehat{\Theta} = \arg\min_{\Theta} &- \sum_{i=1}^{m} \beta_i^T \bar{\phi}_i - \sum_{i<j} \alpha_{ij}^T \bar{\psi}_{ij} + \frac{1}{n} \sum_{l=1}^{n} A(\Theta, \mathbf{x}^{(l)})\\ &+ \frac{\lambda_1}{n} \sum_{i=1}^{m} \|\beta_i\|_2^2 + \frac{\lambda_2}{n} \sum_{i<j} \|\alpha_{ij}\|_2, \qquad (4.21)\end{aligned}$$

where $\bar{\phi}_i = \frac{1}{n} \sum_{l=1}^{n} \mathbf{y}_i^{(l)} \mathbf{x}^{(l)}$, $\bar{\psi}_{ij} = \frac{1}{n} \sum_{l=1}^{n} \mathbf{y}_i^{(l)} \mathbf{y}_j^{(l)} \mathbf{x}^{(l)}$. Note that we have included $A(\Theta, \mathbf{x})$ into (4.6) before the derivation, and thrown away all other terms that are independent of Θ.

Denoting by $\mathcal{L}(\Theta)$ the objective function on the right-hand-side of (4.21). To learn the parameters Θ, a direct gradient-based optimizer is inapplicable due to the non-smooth $\ell_{2,1}$-norm regularizer. In addition, the intractable log-partition function $A(\Theta, \mathbf{x})$ makes the optimization even more complicated. As an alternative, we optimize $\mathcal{L}(\Theta)$ by first dividing the objective into smooth and nonsmooth parts, and then apply the soft thresholding technique. Meanwhile, the mean field approximation is employed to approximate the gradient of $A(\Theta, \mathbf{x})$.

More specifically, we first separate out the smooth part of $\mathcal{L}(\Theta)$ and denote it by $J_s(\Theta)$, i.e.,

$$\begin{aligned} J_s(\Theta) = & -\sum_{i=1}^{m} \beta_i^T \bar{\phi}_i - \sum_{i<j} \alpha_{ij}^T \bar{\psi}_{ij} + \frac{1}{n} \sum_{l=1}^{n} A(\Theta, \mathbf{x}^{(l)}) \\ & + \frac{\lambda_1}{n} \sum_{i=1}^{m} \|\beta_i\|_2^2. \end{aligned} \tag{4.22}$$

Further, according to the mean field approximation described in Section 4.4.1, the gradient of $A(\Theta, \mathbf{x})$ is estimated by replacing the true conditional distribution $p(\mathbf{y}|\mathbf{x})$ with the variational distribution $\widehat{q}(\mathbf{y})$. Hence, we have

$$\left\{ \begin{array}{l} \nabla A_{\beta_i}(\Theta, \mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\mathbf{y}_i \mathbf{x}] \approx \mathbb{E}_{\widehat{q}(\mathbf{y})}[\mathbf{y}_i \mathbf{x}] \\ \nabla A_{\alpha_{ij}}(\Theta, \mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\mathbf{y}_i \mathbf{y}_j \mathbf{x}] \approx \mathbb{E}_{\widehat{q}(\mathbf{y})}[\mathbf{y}_i \mathbf{y}_j \mathbf{x}]. \end{array} \right. \tag{4.23}$$

This results in a simple approximation of the gradient ∇J_s at the k-th iteration $\Theta^{(k)} = \{\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)}\}$ as below

$$\left\{ \begin{array}{l} \nabla J_{s\beta_i}(\Theta^{(k)}) \approx -\bar{\phi}_i + \frac{1}{n} \sum\limits_{l=1}^{n} \widehat{q}(\mathbf{y}_i^{(l)}) \mathbf{x}^{(l)} + \frac{2\lambda_1}{n} \beta_i^{(k)} \\ \nabla J_{s\alpha_{ij}}(\Theta^{(k)}) \approx -\bar{\psi}_{ij} + \frac{1}{n} \sum\limits_{l=1}^{n} \widehat{q}(\mathbf{y}_i^{(l)}) \widehat{q}(\mathbf{y}_j^{(l)}) \mathbf{x}^{(l)}. \end{array} \right. \tag{4.24}$$

Then, a surrogate $J(\Theta)$ of the objective function $\mathcal{L}(\Theta)$ can be obtained by using

$\nabla J_s(\Theta^{(k)})$, i.e.,

$$\begin{aligned}J(\Theta;\Theta^{(k)}) &= J_s(\Theta^{(k)}) \\ &+ \sum_{i=1}^{m}\langle \nabla J_{s\beta_i}(\Theta^{(k)}), \beta_i - \beta_i^{(k)}\rangle + \frac{1}{2\eta}\|\beta_i - \beta_i^{(k)}\|_2^2 \\ &+ \sum_{i<j}\langle \nabla J_{s\alpha_{ij}}(\Theta^{(k)}), \alpha_{ij} - \alpha_{ij}^{(k)}\rangle \\ &+ \frac{1}{2\eta}\|\alpha_{ij} - \alpha_{ij}^{(k)}\|_2^2 + \frac{\lambda_2}{n}\|\alpha_{ij}\|_2. \qquad (4.25)\end{aligned}$$

The parameter η in (4.25) serves as a similar role to the variable updating step size in gradient descent methods. It can be shown that $J(\Theta) \geq \mathcal{L}(\Theta)$ and $J(\Theta^{(k)}) = \mathcal{L}(\Theta^{(k)})$ if $1/\eta$ is larger than the Lipschitz constant of $\nabla J_s(\Theta^{(k)})$. Hence, Θ can be updated by minimizing (4.25), i.e.,

$$\Theta^{(k+1)} = \arg\min_{\Theta} J(\Theta;\Theta^{(k)}), \qquad (4.26)$$

which is solved by

$$\begin{cases} \beta_i^{(k+1)} = \beta_i^{(k)} - \eta\nabla J_{s\beta_i}(\Theta^{(k)}) \\ \alpha_{ij}^{(k+1)} = \mathcal{S}(\alpha_{ij}^{(k)} - \eta\nabla J_{s\alpha_{ij}}(\Theta^{(k)}); \frac{\lambda_2}{n}), \end{cases} \qquad (4.27)$$

where the soft thresholding function is

$$\mathcal{S}(u;\rho) = \begin{cases} (1 - \frac{\rho}{\|u\|_2})u, & \text{if } \|u\|_2 > \rho; \\ 0, & \text{otherwise.} \end{cases} \qquad (4.28)$$

Iteratively applying (4.27) until convergence provides a first-order method for solving (4.21). The pseudo code for this procedure is summarized in Algorithm 3. Note that the gradient descent steps in Algorithm 3 can be speeded up with modern optimization procedures, such as the fast iterative shrinkage thresholding [4].

As a final remark, the conditional graph structure learned by CGL is largely related to the value of hyperparameter λ_2. In general, a larger λ_2, which represents a more peaked Multi-Laplacian prior over $\boldsymbol{\alpha}$, can lead to a sparser conditional structure. As a consequence, it is important to find an appropriate level of sparsity, which can be achieved by resorting to domain knowledge or data-driven cross-validation techniques.

4.5 Experiments

In this section, we evaluate the performance of CGL on the task of multi-label image classification. In particular, all experiments are conducted on three benchmark multi-label image datasets, including MULAN scene (MULANscene)[1], PASCAL VOC 2007 (PASCAL07) [41] and PASCAL VOC 2012 (PASCAL12) [40]. MULAN scene dataset contains 2047 images with 6 labels, and each image is represented by a 294-dimensional feature. PASCAL VOC 2007 dataset consists of 9963 images with 20 labels. For PASCAL VOC 2012, we use public available train and validation subsets which contains 11540 images with 20 labels. As for image features of the latter two datasets, two kinds of feature extractors are employed, i.e., the PHOW (a variant of dense SIFT descriptors extracted at multiple scales) features [13] and the deep CNN (convolutional neural network) features [23, 60]. We extract PHOW features of 3600 dimensions by using the VLFeat implementation [129]. For deep CNN features, we use MatConvNet matlab toolbox [130] and the 'imagenet-vgg-f' model pretrained on ImageNet database [23] to represent each image as a 4096-dimensional feature. The basic information of the datasets is summarized in Table 4.1.

[1] `http://mulan.sourceforge.net/`

Table 4.1: Datasets summary. #images stands for the number of all images, #features stands for the dimension of the features, and #labels stands for the number of labels.

Dataset	#images	#features	#labels
MULANscene	2047	294	6
PASCAL07-PHOW	9963	3600	20
PASCAL07-CNN	9963	4096	20
PASCAL12-PHOW	11540	3600	20
PASCAL12-CNN	11540	4096	20

4.5.1 Label Graph Structure of CGL

To build up an intuition on structure learning of CGL, we employ PASCAL07 with CNN features to visualize the label correlations under different levels of sparsity regularization. In particular, we fix hyperparameter $\lambda_1 = 0.01$ and let λ_2 varies in the range $0.001 \sim 0.1$ to check the label graph evolvement. Since CGL models pairwise label correlations via a parametric linear function, i.e., $\omega_{ij}(\mathbf{x}) = \alpha_{ij}^T \mathbf{x}$, the label graph is actually dependent on features thus unique for each image. To simplify the visualization of so many label graphs, we use the average feature of training images, i.e., $\bar{\mathbf{x}} = \frac{1}{n} \sum_l \mathbf{x}^{(l)}$, and consider the average label graph.

Figure 4.2 presents the graph structure variations as λ_2 increases. From the four label graphs, the number of edges shrinks as λ_2 increases. In addition, the maintained edges are consistent with both semantic co-occurrence (e.g., chair and table) and repulsion (e.g., cat and dog) edges. For co-occurrence, "chair" and "table" often co-appear in the dataset and have large positive correlations, thus the edge weight in the label graph is a large positive value. In contrast, "cat" and "dog" share certain visual similarity, though they seldom co-appear in the dataset. These two terms can be easily treated as conditionally independent by considering label only. However, CGL can successfully capture the repulsion between these two terms, which is represented as a large negative edge weight in the label graph. It is not astonishing since CGL takes both feature and label into account when modeling label correlations.

Figure 4.3 presents the conditional label co-occurrence matrices of three example images from PASCAL07. First of all, the label graph is symmetric and the color of each block represents the correlation strength between two labels. Secondly, for the three different images, the label graphs maintain the same structure. This is not astonishing due to the effect of group sparsity over the pairwise parameters $\{\alpha_{ij}\}$. Note that, the prominent (chair, table) correlation is successfully captured by CGL. Thirdly, the correlation strength of the three images differs from each other. In particular, the (chair, table) correlation strength of the third image (which is in dark red) is much stronger than the first two images (which are in light red). This observation also validates our motivation of incorporating image features during label graph learning.

4.5.2 Comparison Methods and Measures

We compare CGL with the binary relevance (BR) method and six state-of-the-art multi-label classification methods. Here we use logistic regression to implement BR method which is also named as the independent logistic regressions (ILRs) method. Moreover, six state-of-the-art multi-label classification methods - instance-based learning by logistic regression (IBLR) [28], multi-label k-nearest neighbor (MLKNN) [152], classifier chains (CC) [108], maximum margin output coding (MMOC) [155], probabilistic label enhancement model (PLEM) [77] and clique generating machine (CGM) [121] were also employed for comparison study. Note that ILRs can be regarded as the basic baseline and other methods represent state-of-the-arts. In our experiments, LIBlinear [42] ℓ_2-regularized logistic regression is employed to build binary classifiers for ILRs. Based on ILRs, we implement PLEM by ourselves. As for other methods, we use publicly available codes in MEKA [1] and the authors' homepages [2] [3].

[1] http://meka.sourceforge.net/

[2] http://www.cs.cmu.edu/~yizhang1/

[3] http://www.tanmingkui.com/cgm.html

Table 4.2: Multi-label image classification performance comparison on MULANscene via 5-fold cross validation

Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
MULANscene	ILRs	0.117±0.006	0.495±0.022	0.592±0.016	0.622±0.014	0.677±0.016	0.669±0.014
	IBLR	**0.085±0.004**	0.358±0.016	0.677±0.018	0.689±0.019	**0.747±0.010**	**0.738±0.014**
	MLKNN	0.086±0.003	0.374±0.015	0.668±0.018	0.682±0.019	0.742±0.013	0.734±0.012
	CC	0.104±0.005	**0.346±0.015**	0.696±0.015	0.710±0.015	0.716±0.018	0.706±0.014
	MMOC	0.126±0.017	0.401±0.046	0.629±0.049	0.639±0.050	0.680±0.031	0.638±0.049
	PLEM	0.096±0.005	0.423±0.010	0.627±0.011	0.644±0.012	0.713±0.017	0.704±0.014
	CGM	0.096±0.004	0.390±0.016	0.647±0.016	0.659±0.016	0.717±0.011	0.708±0.012
	CGL	0.096±0.006	0.347±0.019	**0.705±0.019**	**0.724±0.020**	0.745±0.015	0.731±0.018

Table 4.3: Multi-label image classification performance comparison on PASCAL07 via 5-fold cross validation

PASCAL07-PHOW	ILRs	0.093±0.001	0.878±0.007	0.294±0.008	0.360±0.009	0.332±0.008	0.404±0.007
	IBLR	0.066±0.001	0.832±0.003	0.270±0.005	0.308±0.006	0.258±0.007	0.408±0.009
	MLKNN	0.066±0.001	0.839±0.006	0.256±0.007	0.291±0.008	0.235±0.006	0.392±0.007
	CC	0.091±0.000	0.845±0.010	0.318±0.005	0.379±0.003	0.348±0.004	0.417±0.001
	MMOC	**0.065±0.001**	0.850±0.003	0.259±0.009	0.299±0.011	0.206±0.007	0.392±0.012
	PLEM	0.066±0.001	0.800±0.005	0.319±0.009	0.362±0.010	0.324±0.013	0.445±0.011
	CGM	0.073±0.002	0.819±0.011	0.327±0.010	0.381±0.010	0.359±0.014	0.450±0.011
	CGL	0.070±0.002	**0.742±0.010**	**0.386±0.011**	**0.433±0.011**	**0.371±0.012**	**0.475±0.014**
PASCAL07-CNN	ILRs	0.046±0.001	0.574±0.011	0.610±0.010	0.673±0.009	0.651±0.004	0.688±0.007
	IBLR	0.043±0.001	0.554±0.011	0.597±0.014	0.649±0.015	0.621±0.007	0.682±0.010
	MLKNN	0.043±0.001	0.557±0.010	0.585±0.014	0.635±0.015	0.613±0.006	0.668±0.011
	CC	0.051±0.001	0.586±0.008	0.602±0.008	0.668±0.008	0.635±0.009	0.669±0.008
	MMOC	**0.037±0.000**	0.512±0.008	0.634±0.009	0.684±0.009	0.663±0.005	0.719±0.004
	PLEM	0.045±0.001	0.555±0.011	0.619±0.009	0.678±0.009	0.654±0.008	0.694±0.008
	CGM	0.044±0.001	0.552±0.011	0.628±0.009	0.689±0.009	0.661±0.006	0.702±0.009
	CGL	0.040±0.001	**0.480±0.010**	**0.676±0.009**	**0.730±0.009**	**0.680±0.007**	**0.726±0.008**

Table 4.4: Multi-label image classification performance on PASCAL12 comparison via 5-fold cross validation

PASCAL12-PHOW	ILRs	0.100±0.001	0.891±0.009	0.269±0.007	0.333±0.008	0.324±0.008	0.370±0.005
	IBLR	0.068±0.001	0.869±0.009	0.219±0.005	0.252±0.003	0.253±0.007	0.345±0.005
	MLKNN	0.069±0.001	0.883±0.008	0.191±0.006	0.218±0.005	0.213±0.007	0.306±0.006
	CC	0.097±0.001	0.862±0.012	0.291±0.010	0.350±0.010	0.340±0.007	0.380±0.006
	MMOC	**0.067±0.001**	0.865±0.003	0.227±0.005	0.262±0.007	0.200±0.007	0.346±0.004
	PLEM	0.068±0.001	0.823±0.009	0.286±0.009	0.325±0.009	0.326±0.012	0.405±0.008
	CGM	0.076±0.002	0.836±0.007	0.302±0.009	0.352±0.010	0.361±0.015	0.417±0.011
	CGL	0.076±0.001	**0.762±0.006**	**0.365±0.007**	**0.413±0.007**	**0.380±0.007**	**0.442±0.005**
PASCAL12-CNN	ILRs	0.051±0.001	0.613±0.002	0.581±0.005	0.649±0.006	0.638±0.005	0.658±0.005
	IBLR	0.045±0.001	0.574±0.006	0.575±0.009	0.627±0.010	0.613±0.008	0.657±0.006
	MLKNN	0.045±0.002	0.575±0.012	0.566±0.015	0.616±0.017	0.604±0.011	0.645±0.013
	CC	0.055±0.001	0.615±0.010	0.579±0.009	0.647±0.010	0.623±0.005	0.643±0.007
	MMOC	**0.039±0.001**	0.525±0.005	0.619±0.006	0.669±0.007	0.659±0.004	0.699±0.005
	PLEM	0.049±0.001	0.592±0.006	0.590±0.003	0.653±0.003	0.639±0.004	0.664±0.004
	CGM	0.047±0.001	0.583±0.006	0.603±0.006	0.666±0.007	0.650±0.005	0.677±0.006
	CGL	0.042±0.001	**0.498±0.010**	**0.661±0.005**	**0.717±0.006**	**0.677±0.004**	**0.707±0.003**

We use six widely accepted performance criteria to evaluate all the methods, including four example based measures (Hamming loss, zero-one loss, accuracy and F1-score) and two label based measures (Macro-F1 and Micro-F1). In general, example based measures encourage the importance of performing well on each example, the Macro-F1 score is more influenced by the performance on rare categories, and the Micro-F1 score tend to be dominated by the performance on common categories. More details of these evaluation measures can be found in [87, 144]. It is worth mentioning that, PLEM, CGM and our method solve MPE inference problem for label prediction (each predicted label is either 0 or 1 thus containing no ranking information). As a result, ranking based measures like mean average precision (mAP) are not suitable for these methods. In addition, all the methods are compared by 5-fold cross validation on each dataset. And the mean and standard deviation are reported for each criterion.

4.5.3 Results and Discussion

Tables 4.2, 4.3, 4.4 summarize the experimental results on MULANscene, PASCAL07 and PASCAL12 of all eight algorithms evaluated by the six measures. Except for Hamming loss, CGL achieves better or comparable results on all datasets with different types of feature. This is because Hamming loss treats the prediction of each label individually. However, CGL performs significantly better than other methods on PASCAL07 and PASCAL12 in terms of the other five measures. Especially in terms of accuracy and F1-score, CGL performs the best on all datasets. It is interesting that these two measures encourage good performance on each example. CGL's outstanding performance on accuracy and F1-score confirms our motivation of exploiting conditional label correlations, which enables example based label graph. In the following, we present a more detailed comparison between CGL and the four different categories of multi-label classification methods.

We first compare CGL with problem transformation methods (ILRs and CC). We observe that both CGL and CC outperforms ILRs which validates the improvements obtained by exploiting label correlations for multi-label classification. However, CC has to incrementally conduct training and prediction thus is not scalable to large label space.

Secondly, CGL shows better performance than algorithm adaptation methods (IBLR and MLKNN). Both IBLR and MLKNN adopt a local approach to adjust label prediction performance for each image instance. However, such lazy learners can be very inefficient when making predictions especially when the training database is large.

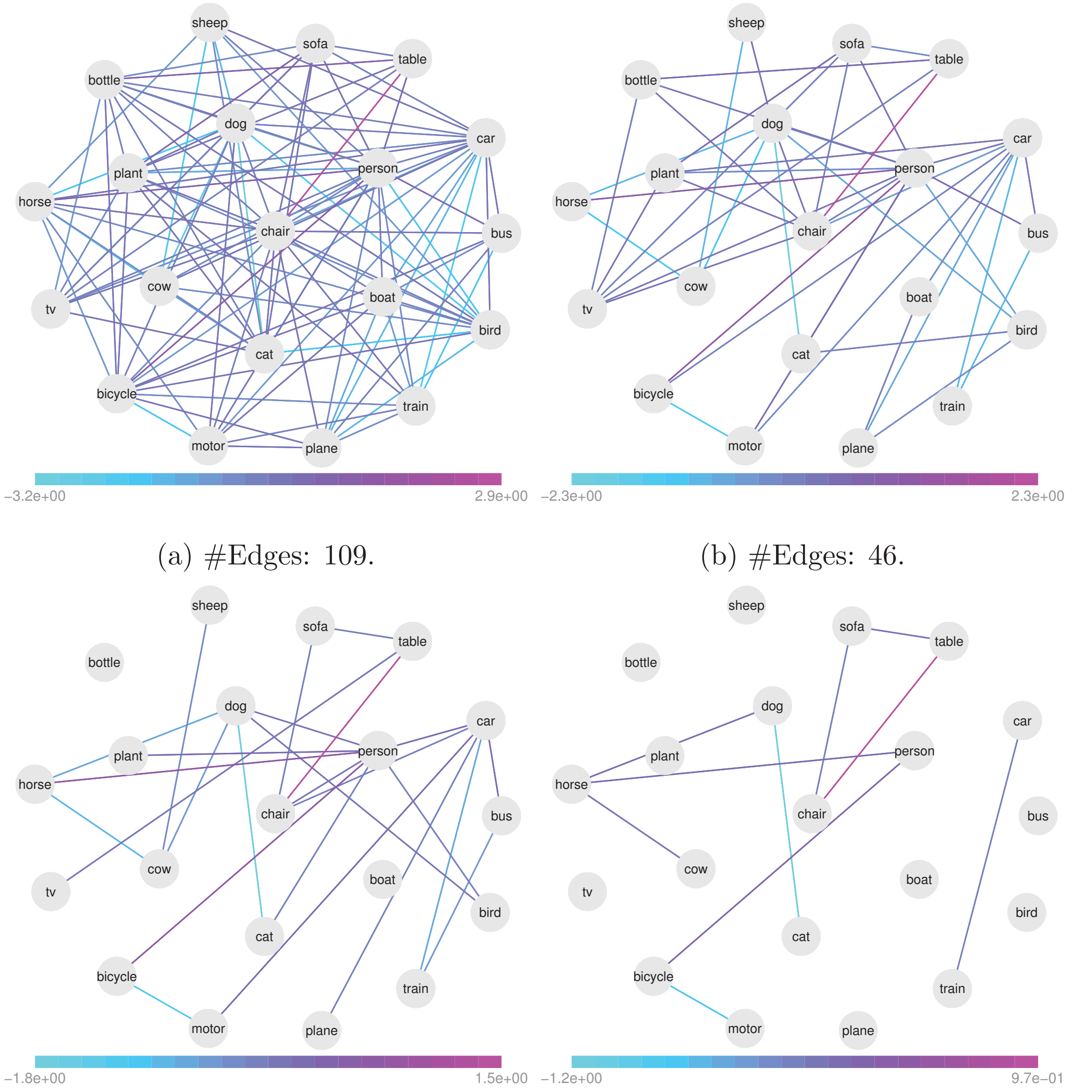
Thirdly, CGL outperforms the label space dimension reduction algorithm MMOC. Though MMOC obtained good performance on PASCAL07-CNN and PASCAL12-CNN, the training of output codes is time-consuming. In addition, MMOC is sensitive to the features at hand since its performance degrades more than other methods when PHOW feature is utilized instead of CNN.

Finally, we compare three structure learning based methods (PLEM, CGM and CGL). One can observe that both CGL and CGM performs better than PLEM on all datasets. This is because PLEM learns the label graph based on label statistics without using the features. On the other hand, CGM learns a shared label graph across all images which lacks flexibility. In contrast, CGL exploits conditional label correlations that are adaptive to different images.

To investigate how CGL exploits label correlations, we present the performance variation of CGL versus the hyperparamter λ_2 on MULANscene and PASCAL07-CNN. We use the same setting in Section 4.5.1 by letting $\lambda_1 = 0.01$ and λ_2 range from 0.001 to 0.1. The results are shown in Figures 4.4 and 4.5. To make the performance variation easier to understand, we also provide the curve of #Edges versus λ_2 in Figures 4.4a and 4.5a. According to the two curves, larger λ_2 encourages graph sparsity which leads to fewer edges. As for the performance curves, we can draw several conclusions. First, the performance almost keeps stable when λ is larger than some value since few label correlations have been utilized. Second, utilizing more relevant label correlations can improve the performance. However, adding too many label correlations (especially irrelevant ones) may impair the performance due to overfitting issues.

4.6 Summary

A conditional structure learning approach has been developed for multi-label image classification. Our proposed conditional graphical lasso framework offers a principled way to model label correlations by jointly considering image features and labels. In addition, our proposed framework is provided with a graceful Bayesian interpretation. The multi-label prediction task is formulated into an inference problem which is handled via an efficient mean field approximate procedure. And the learning problem is efficiently solved by a tailored proximal gradient algorithm. Empirical evaluations confirmed the effectiveness of our method and showed its superiority over other state-of-the-art multi-label classification algorithms.



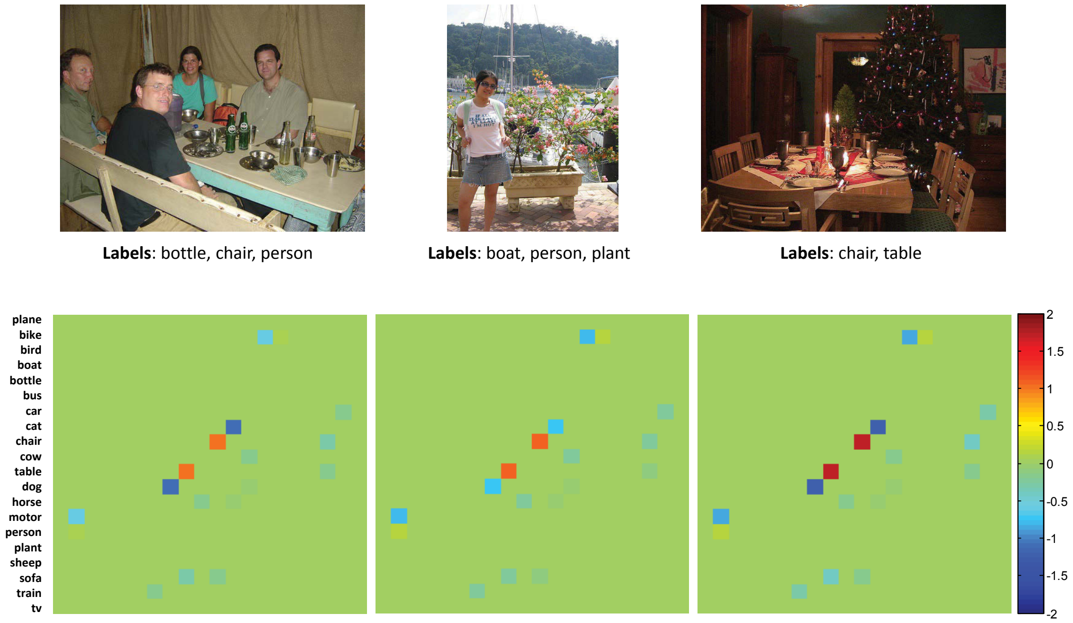
(a) #Edges: 109.

(b) #Edges: 46.

(c) #Edges: 25.

(d) #Edges: 10.

Figure 4.2: Illustration of the CGL label graphs learned from PASCAL07-CNN.

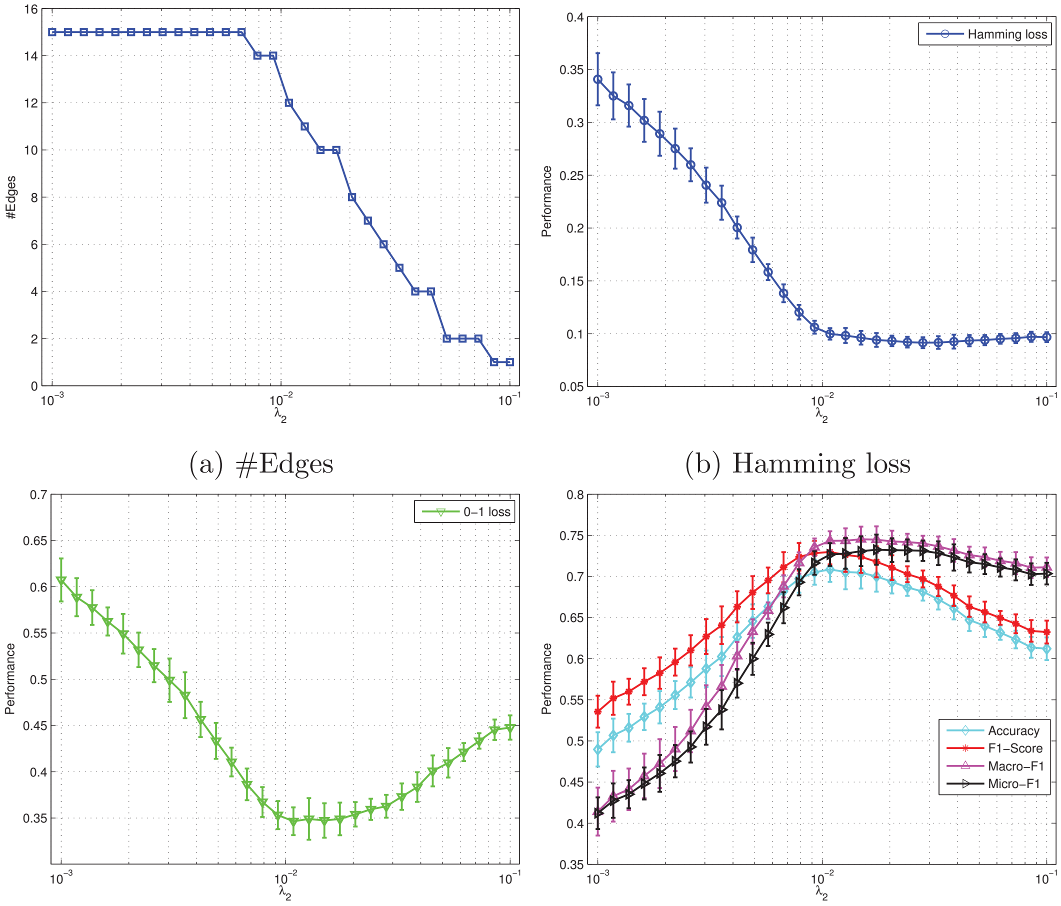


Labels: bottle, chair, person

Labels: boat, person, plant

Labels: chair, table

Figure 4.3: Illustration of the CGL label graphs for test example images in PASCAL07-CNN.



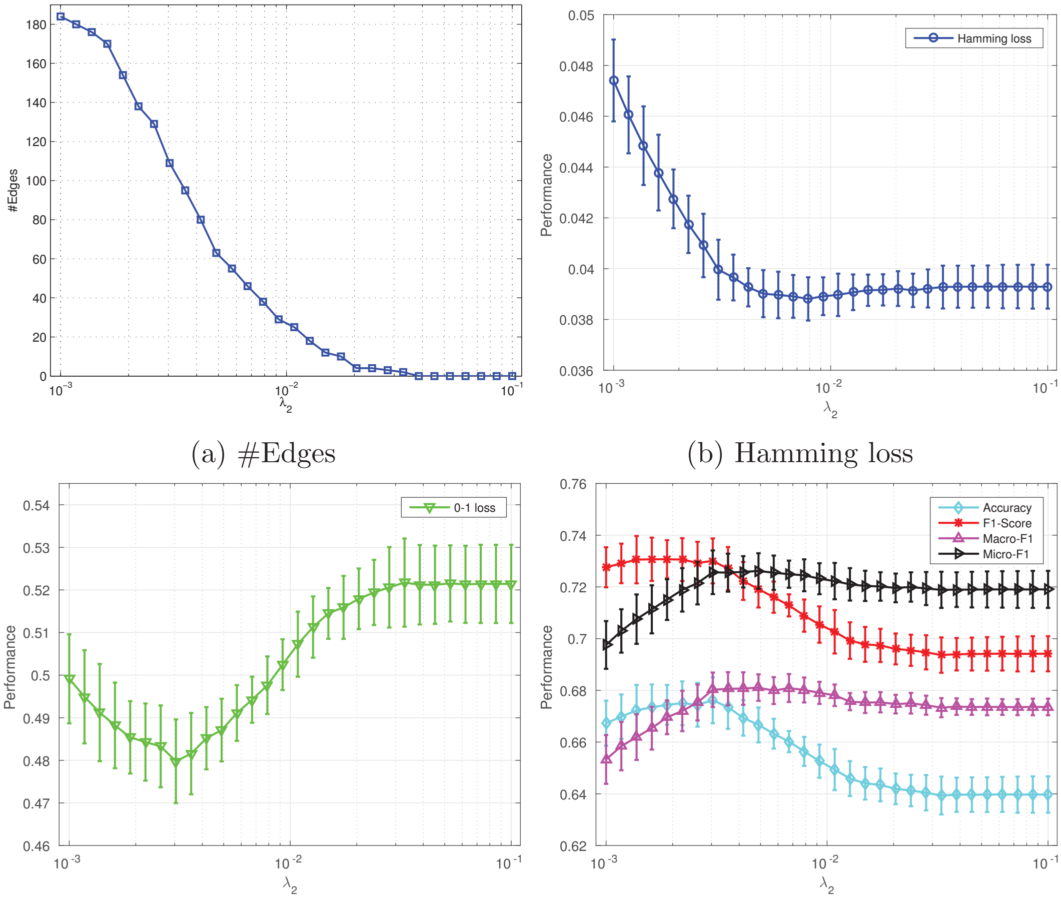
(a) #Edges

(b) Hamming loss

(c) 0-1 loss

(d) Other measures

Figure 4.4: Performance variation of CGL versus the hyperparameter λ_2 on MULANscene.



(a) #Edges

(b) Hamming loss

(c) 0-1 loss

(d) Other measures

Figure 4.5: Performance variation of CGL versus the hyperparameter λ_2 on PASCAL07-CNN.

Chapter 5

Conclusions

5.1 Summary of This Thesis

In this thesis, we studied how to learn sparse graphical models to exploit mixed-attribute and conditional label correlations to handle data restoration and multi-label classification tasks. To this end, we have the following conclusions:

- Current existing data restoration methods are either limited to a particular task, or incapable of handling mixed-attribute data. The proposed random mixed field (RMF) model derives from general properties of mixed graphical models. RMF provides an elegant unified framework for exploiting mixed-attribute correlations among data. Based upon a regularized maximum pseudo likelihood estimation procedure, RMF learning can be achieved by using proximal gradient descent method. Meanwhile, by using a structured mean-field approach, RMF inference is effective for data denoising and imputation tasks simultaneously.

- Learning conditional label correlations is beneficial to improve the performance of multi-label classification. The proposed elastic-net correlated logistic (CorrLog) model explicitly handles label correlations by scalar parameters. By inducing elastic-net regularization over these parameters, CorrLog is able to obtain an interpretable sparse label graph. CorrLog learning is also implemented according to the regularized maximum pseudo likelihood estimation. Meanwhile, CorrLog inference is based on the message passing algorithm which is effective for multi-label classification according to empirical evaluations on music annotation and image classification.

- The proposed conditional graphical lasso (CGL) implicitly handles label correlations via parametric functions of input features. CGL provides a unified Bayesian framework for structure and parameter learning conditioned on input features. The learned label graph has good semantic interpretations and benefits multi-label classification. By applying the maximum a posterior methodology, CGL learning is efficient via a proximal gradient procedure. Besides, CGL inference relies on a mean-field approach which is effective for multi-label classification according to empirical evaluations on benchmark image classification datasets.

5.2 Future Works

In this thesis, we studied sparse graphical models in terms of parametric and attraction-based settings. Future efforts may extend sparse graphical models in three directions by considering non-parametric (or semi-parametric) tools such as kernel [114] and copula [39], by leveraging repulsion such as determinant point process [62] and submodular methods [3], and by compensating uncertainty via Bayesian methodology [47].

- Parametric methods are usually tied with the parametric function form being used thus inflexible to model complex data and its model capacity cannot improve as we provide more data. In contrast, the non-parametric approach offers a natural recipe for handling complex data and scalability of model capacity. For instance, copula provides a graceful semi-parametric approach to design sparse graphical models [81]. In general, the copula approach explicitly models correlations between independent univariate distributions via a dependency matrix. By combining copula, sparsity and exponential family distributions, we could design more powerful, flexible and manageable sparse graphical models to discover attribute dependence among heterogenous data.

- Currently, most of the sparse graphical models are based on the attraction assumption. However, real-world data may also contain repulsive relation which violate the attraction assumption. A recent progress on submodular graphical models [38] validated the possibility of compromising attraction and repulsion in a unified framework.

- The third direction is a widely employed methodology in extending non-Bayesian graphical models. In particular, a Bayesian compensation of uncertainty over model parameters could provide more reasonable posterior estimate of graph structure an parameters. A recent progress along this direction is the Bayesian graphical lasso [137] which investigated differences between the posterior mean estimate and the posterior mode estimate (obtained by graphical lasso) of the graph structure. Besides, we can move further by considering Bayesian sparse non-parametric (or submodular) graphical models.

References

[1] D. Angluin and P. Laird, “Learning from noisy examples,” *Mach. Learn.*, vol. 2, no. 4, pp. 343–370, 1988. 10

[2] J. Ashford and R. Sowden, “Multi-variate probit analysis,” *Biometrics*, vol. 26, pp. 535–546, 1970. 52

[3] F. Bach *et al.*, “Learning with submodular functions: A convex optimization perspective,” *Foundations and Trends® in Machine Learning*, vol. 6, no. 2-3, pp. 145–373, 2013. 107

[4] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM J. Img. Sci.*, vol. 2, no. 1, pp. 183–202, 2009. 57, 94

[5] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, 2002. 14, 82

[6] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 192–236, 1974. 27

[7] ——, “Statistical Analysis of Non-Lattice Data,” *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 24, no. 3, pp. 179–195, 1975. 54

[8] C. Bhattacharyya, K. Pannagadatta, and A. J. Smola, “A second order cone programming formulation for classifying missing data,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 153–160. 10

[9] J. Bi and T. Zhang, “Support vector classification with input data uncertainty,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 161–168. 10

[10] W. Bian, B. Xie, and D. Tao, “Corrlog: Correlated logistic models for joint prediction of multiple labels,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2012, pp. 109–117. 48

[11] D. Bickson, “Gaussian belief propagation: Theory and aplication,” *arXiv preprint arXiv:0811.2518*, 2008. 26

[12] C. M. Bishop, *Pattern recognition and machine learning.* Springer, 2006. 1, 25, 58, 89

[13] A. Bosch, A. Zisserman, and X. Munoz, “Image classification using random forests and ferns,” in *Proc. IEEE Int. Conf. Comput. Vis.* IEEE, 2007, pp. 1–8. 15, 72, 82, 95

[14] O. Bousquet and A. Elisseeff, “Stability and generalization,” *J. Mach. Learn. Res.*, vol. 2, pp. 499–526, 2002. 58, 64, 66

[15] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, “Learning multi-label scene classification,” *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004. 14, 82

[16] J. K. Bradley and C. Guestrin, “Learning tree conditional random fields,” in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 127–134. 83

[17] L. Breiman and J. H. Friedman, “Predicting multivariate responses in multiple linear regression (with discussion),” *Journal of the Royal Statistical*

Society: Series B (Statistical Methodology), vol. 59, no. 1, pp. 3–54, 1997. 49, 51

[18] G. Bresler, D. Gamarnik, and D. Shah, "Structure learning of antiferromagnetic ising models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2852–2860. 4

[19] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. IEEE, 2005, pp. 60–65. 8, 18

[20] ——, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005. 18

[21] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Mining hidden community in heterogeneous social networks," in *International Workshop on Link Discovery.* ACM, 2005, pp. 58–65. 7

[22] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *J. Mach. Learn. Res.*, vol. 7, pp. 31–54, 2006. 49, 85

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.* BMVA Press, 2014, pp. 6:1–6:12. 15, 72, 82, 95

[24] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller, "Max-margin classification of incomplete data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 233–240. 11

[25] ——, "Max-margin classification of data with absent features," *J. Mach. Learn. Res.*, vol. 9, pp. 1–21, 2008. 11

[26] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1529–1537. 85

[27] J. Cheng, E. Levina, and J. Zhu, "High-dimensional mixed graphical models," *arXiv preprint arXiv:1304.2810*, 2013. 5, 20

[28] W. Cheng and E. Hüllermeier, "Combining instance-based learning and logistic regression for multilabel classification," *Mach. Learn.*, vol. 76, no. 2-3, pp. 211–225, 2009. 47, 49, 74, 85, 97

[29] W. Cheng, E. Hüllermeier, and K. J. Dembczynski, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 279–286. 50, 75, 84

[30] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 462–467, 1968. 83

[31] E. Coviello, L. Barrington, A. B. Chan, and G. Lanckriet, "Automatic music tagging with time series models," in *Proc. Int. Society Music Inf. Retrieval Conf.*, 2010, pp. 81–86. xvi, 14, 69, 70, 71

[32] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007. 8

[33] C. Dahinden, G. Parmigiani, M. C. Emerick, and P. Bühlmann, "Penalized likelihood for sparse contingency tables with an application to full-length cdna libraries," *BMC Bioinformatics*, vol. 8, no. 1, p. 476, 2007. 3

[34] O. Dekel, O. Shamir, and L. Xiao, "Learning to classify with missing and corrupted features," *Mach. Learn.*, vol. 81, no. 2, pp. 149–178, 2010. 11, 32

[35] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence in multi-label classification," in *Proc. Int. Conf. Mach. Learn. Workshop on Learning from Multi-label Data*, 2010, pp. 5–13. 12, 47, 48, 49, 51

[36] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence and loss minimization in multi-label classification," *Mach. Learn.*, vol. 88, no. 1-2, pp. 5–45, 2012. 12, 47, 48

[37] Z. Ding, M. Shao, and Y. Fu, "Latent low-rank transfer subspace learning for missing modality recognition," in *AAAI Conf. Artif. Intell.*, 2014, pp. 1192–1198. 20

[38] J. Djolonga, S. Tschiatschek, and A. Krause, "Variational inference in mixed probabilistic submodular models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1759–1767. 108

[39] G. Elidan, "Copulas in machine learning," in *Copulae in mathematical and quantitative finance.* Springer, 2013, pp. 39–60. 107

[40] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015. 72, 95

[41] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. 72, 95

[42] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008. 74, 97

[43] S. Feng, R. Manmatha, and V. Lavrenko, “Multiple bernoulli relevance models for image and video annotation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. IEEE, 2004, pp. II–1002. 14, 82

[44] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008. 3, 27

[45] M. Frydenberg and S. L. Lauritzen, “Decomposition of maximum likelihood in mixed graphical interaction models,” *Biometrika*, vol. 76, no. 3, pp. 539–555, 1989. 5

[46] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, “A survey of audio-based music classification and annotation,” *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, 2011. 13

[47] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis.* Chapman & Hall/CRC Boca Raton, FL, USA, 2014, vol. 2. 107

[48] Z. Ghahramani, G. E. Hinton *et al.*, “The em algorithm for mixtures of factor analyzers,” Technical Report CRG-TR-96-1, University of Toronto, Tech. Rep., 1996. 8

[49] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an em approach,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 120–127. 8, 9, 19

[50] E. Gibaja and S. Ventura, “A tutorial on multilabel learning,” *ACM Comput. Surv.*, vol. 47, no. 3, p. 52, 2015. 13, 84

[51] M. D. Hoffman, D. M. Blei, and P. R. Cook, “Easy as cba: A simple probabilistic model for tagging music,” in *Proc. Int. Society Music Inf. Retrieval Conf.* xvi, 13, 14, 69, 70, 71

[52] F. Honghai, C. Guoshun, Y. Cheng, Y. Bingru, and C. Yumei, "A svm regression based approach to filling in missing values," in *Knowledge-Based Intelligent Information and Engineering Systems.* Springer, 2005, pp. 581–587. 9, 19

[53] D. Hsu, S. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing." in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 22, 2009, pp. 772–780. 47, 49, 85

[54] X. Huang, L. Shi, and J. A. Suykens, "Support vector machine classifier with pinball loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 984–997, 2014. 10

[55] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2010, pp. 3304–3311. 15, 82

[56] S. Ji, L. Tang, S. Yu, and J. Ye, "A shared-subspace learning framework for multi-label classification," *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 2, p. 8, 2010. 85

[57] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for dna microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005. 9, 19

[58] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques.* MIT press, 2009. 1, 82

[59] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label relieff and f-statistic feature selections for image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 2352–2359. 85

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105. 15, 72, 82, 95

[61] J. Kubica and A. W. Moore, “Probabilistic noise identification and data cleaning,” in *IEEE Int. Conf. Data Min.*, 2003, pp. 131–138. 8

[62] A. Kulesza, B. Taskar *et al.*, “Determinantal point processes for machine learning,” *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012. 107

[63] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. Int. Conf. Mach. Learn.* ACM, 2001, pp. 282–289. 15, 50, 52, 87

[64] S. L. Lauritzen, “Propagation of probabilities, means, and variances in mixed graphical association models,” *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1098–1108, 1992. 5

[65] ——, *Graphical models.* Oxford University Press, 1996. 5

[66] S. L. Lauritzen, A. H. Andersen, D. Edwards, K. G. Jöreskog, and S. Johansen, “Mixed graphical association models [with discussion and reply],” *Scandinavian Journal of Statistics*, pp. 273–306, 1989. 5

[67] S. L. Lauritzen and N. Wermuth, “Graphical models for associations between variables, some of which are qualitative and some quantitative,” *Annals of Statistics*, pp. 31–57, 1989. 5, 20

[68] J. Lee and T. Hastie, “Structure learning of mixed graphical models,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2013, pp. 388–396. 5, 20, 27, 83

[69] S.-I. Lee, V. Ganapathi, and D. Koller, "Efficient structure learning of markov networks using L_1-regularization," in *Proc. Adv. Neural Inf. Process. Syst.* MIT Press, 2006, pp. 817–824. 3

[70] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: A study of fuzzy k-means clustering method," in *Rough Sets and Current Trends in Computing.* Springer, 2004, pp. 573–579. 9

[71] H.-X. Li, J.-L. Yang, G. Zhang, and B. Fan, "Probabilistic support vector machines for classification of noise affected data," *Information Sciences*, vol. 221, pp. 60–71, 2013. 10

[72] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1378–1386. 15, 82

[73] Q. Li, T. Li, S. Zhu, and C. Kambhamettu, "Improving medical/biological data classification performance by wavelet preprocessing," in *IEEE Int. Conf. Data Min.* IEEE, 2002, pp. 657–660. 7, 18

[74] Q. Li, W. Bian, R. Y. D. Xu, J. You, and D. Tao, "Random mixed field model for mixed-attribute data restoration," in *AAAI Conf. Artif. Intell.*, 2016, pp. 1244–1250. 16

[75] Q. Li, M. Qiao, W. Bian, and D. Tao, "Conditional graphical lasso for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2977–2986. 16

[76] Q. Li, B. Xie, J. You, B. Wei, and D. Tao, "Correlated logistic model with elastic net regularization for multilabel image classification," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3801–3813, 2016. 16

[77] X. Li, F. Zhao, and Y. Guo, "Multi-label image classification with a probabilistic label enhancement model," in *Proc. Conf. Uncertain. Artif. Intell.*, 2014, pp. 430–439. 15, 83, 97

[78] W. Lian, P. Rai, E. Salazar, and L. Carin, "Integrating features and similarities: Flexible models for heterogeneous multiview data," in *AAAI Conf. Artif. Intell.*, 2015, pp. 2757–2763. 20

[79] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml 6, 8, 18, 32

[80] R. J. Little and D. B. Rubin, *Statistical analysis with missing data.* John Wiley & Sons, 2014. 6

[81] H. Liu, F. Han, M. Yuan, J. Lafferty, L. Wasserman *et al.*, "High-dimensional semiparametric gaussian copula graphical models," *Annals of Statistics*, vol. 40, no. 4, pp. 2293–2326, 2012. 107

[82] P.-L. Loh, M. J. Wainwright *et al.*, "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses," *Annals of Statistics*, vol. 41, no. 6, pp. 3022–3049, 2013. 4

[83] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2355–2368, 2015. 14

[84] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 523–536, 2013. 14

[85] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 709–722, 2013. 14

[86] L. Maaten, M. Chen, S. Tyree, and K. Q. Weinberger, “Learning with marginalized corrupted features,” in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 410–418. 11, 32

[87] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, “An extensive experimental comparison of methods for multi-label learning,” *Pattern Recognit.*, vol. 45, no. 9, pp. 3084–3104, 2012. 75, 99

[88] O. Maimon and L. Rokach, “Data mining and knowledge discovery handbook,” 2010. 18

[89] B. M. Marlin, “Missing data problems in machine learning,” Ph.D. dissertation, University of Toronto, 2008. 10

[90] X. Mei and H. Ling, “Robust visual tracking using ℓ_1 minimization,” in *Proc. IEEE Int. Conf. Comput. Vis.* IEEE, 2009, pp. 1436–1443. 14

[91] N. Meinshausen and P. Bühlmann, “High-dimensional graphs and variable selection with the lasso,” *Annals of Statistics*, pp. 1436–1462, 2006. 3, 83

[92] R. Miotto, L. Barrington, and G. R. Lanckriet, “Improving auto-tagging by modeling semantic co-occurrences.” in *Proc. Int. Society Music Inf. Retrieval Conf.*, 2010, pp. 297–302. xvi, 14, 69, 70, 71

[93] K. P. Murphy, Y. Weiss, and M. I. Jordan, “Loopy belief propagation for approximate inference: An empirical study,” in *Proc. Conf. Uncertain. Artif. Intell.* Morgan Kaufmann Publishers Inc., 1999, pp. 467–475. 26

[94] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000. 8

[95] R. Neal, "Bayesian mixture modelling by monte carlo simulation," Technical Report CRG–TR–91–2, Computer Science, Univ. of Toronto, Tech. Rep., 1991. 8

[96] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, ser. Wiley-Interscience Series in Discrete Mathematics. New York, USA: John Wiley & Sons, 1983. 57

[97] S. R. Ness, A. Theocharis, G. Tzanetakis, and L. G. Martins, "Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs," in *Proc. ACM Int. Conf. Multimedia.* ACM, 2009, pp. 705–708. 14, 71

[98] S. Oba, M.-a. Sato, I. Takemasa, M. Monden, K.-i. Matsubara, and S. Ishii, "A bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, no. 16, pp. 2088–2096, 2003. 9, 19

[99] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001. 15, 72, 82

[100] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 203–228, 2006. 7

[101] K. Pelckmans, J. De Brabanter, J. A. Suykens, and B. De Moor, "Handling missing values in support vector machine classifiers," *Neural Networks*, vol. 18, no. 5, pp. 684–692, 2005. 11

[102] J. Petterson and T. S. Caetano, "Reverse multi-label learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1912–1920. 47

[103] J. Quinlan, "Learning from noisy data," in *International Machine Learning Workshop.* Citeseer, 1983, pp. 58–64. 10

[104] S. Raman, T. J. Fuchs, P. J. Wild, E. Dahl, and V. Roth, “The bayesian group-lasso for analyzing contingency tables,” in *Proc. Int. Conf. Mach. Learn.* ACM, 2009, pp. 881–888. 88

[105] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty, “High-dimensional ising model selection using ℓ_1-regularized logistic regression,” *Annals of Statistics*, vol. 38, no. 3, pp. 1287–1319, 2010. 52, 54, 83, 86

[106] N. S. Razavian, H. Kamisetty, and C. J. Langmead, “The von mises graphical model: structure learning,” *Carnegie Mellon University School of Computer Science Technical Report*, 2011. 3

[107] J. Read, “A Pruned Problem Transformation Method for Multi-label classification,” in *Proc. New Zealand Computer Science Research Student Conference*, 2008, pp. 143–150. 49, 85

[108] J. Read, B. Pfahringer, G. Holmes, and E. Frank, “Classifier chains for multi-label classification,” *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, 2011. 50, 74, 75, 84, 97

[109] A. Rostamizadeh, A. Agarwal, and P. L. Bartlett, “Learning with missing features.” in *Proc. Conf. Uncertain. Artif. Intell.*, 2011, pp. 635–642. 11

[110] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection.* John Wiley & Sons, 2005, vol. 589. 6

[111] M. Saar-Tsechansky and F. Provost, “Handling missing values when applying classification models,” *J. Mach. Learn. Res.*, vol. 8, pp. 1625–1657, 2007. 27, 32

[112] J. C. Schlimmer and R. H. Granger Jr, “Incremental learning from noisy data,” *Mach. Learn.*, vol. 1, no. 3, pp. 317–354, 1986. 10

[113] T. Schneider, "Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values," *Journal of Climate*, vol. 14, no. 5, pp. 853–871, 2001. 9, 19

[114] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002. 107

[115] A. J. Smola, S. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," p. 325, 2005. 11

[116] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 174–184, 2010. 14

[117] D. J. Stekhoven and P. Bühlmann, "Missforest-non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012. 9, 19

[118] F. Sun, J. Tang, H. Li, G.-J. Qi, and T. S. Huang, "Multi-label image categorization with sparse factor representation," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1028–1037, 2014. 14

[119] C. Sutton and A. McCallum, "Piecewise pseudolikelihood for efficient training of conditional random fields," in *Proc. Int. Conf. Mach. Learn.* ACM, 2007, pp. 863–870. 54

[120] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012. 50, 85

[121] M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang, "Learning graph structure for multi-label image classification via clique generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4100–4109. 15, 77, 83, 97

[122] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999. 8

[123] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999. 8

[124] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001. 9, 18, 19, 35

[125] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010, pp. 667–685. 13, 47, 49, 84, 85

[126] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. Eur. Conf. Mach. Learn.* Springer, 2007, pp. 406–417. 49, 75, 85

[127] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Towards musical query-by-semantic-description using the cal500 data set," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* ACM, 2007, pp. 439–446. 69

[128] ——, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 467–476, 2008. xvi, 14, 69, 71

[129] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia.* ACM, 2010, pp. 1469–1472. 72, 95

[130] A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in *Proc. ACM Int. Conf. Multimedia.* ACM, 2015, pp. 689–692. 72, 95

[131] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. IEEE, 2001, pp. I–I. 14

[132] N. Vlassis and J. Verbeek, “Gaussian mixture learning from noisy data,” 2004. 8

[133] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008. 1

[134] M. J. Wainwright, J. D. Lafferty, and P. K. Ravikumar, “High-dimensional graphical model selection using ℓ_1-regularized logistic regression,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1465–1472. 3, 4

[135] C. Wang, C.-H. Chi, W. Zhou, and R. Wong, “Coupled interdependent attribute analysis on mixed data,” in *AAAI Conf. Artif. Intell.*, 2015, pp. 1861–1867. 20

[136] G. Wang, D. Forsyth, and D. Hoiem, “Improved object categorization and detection using comparative object similarity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2442–2453, 2013. 14

[137] H. Wang *et al.*, “Bayesian graphical lasso models and efficient posterior computation,” *Bayesian Analysis*, vol. 7, no. 4, pp. 867–886, 2012. 108

[138] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, “Video tracking using learned hierarchical features,” *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1424–1435, 2015. 14

[139] C. Xu, T. Liu, D. Tao, and C. Xu, "Local rademacher complexity for multi-label learning," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1495–1507, 2016. 14

[140] H. Xu, C. Caramanis, and S. Mannor, "Sparse algorithms are not stable: A no-free-lunch theorem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 187–193, 2012. 67

[141] E. Yang, G. Allen, Z. Liu, and P. K. Ravikumar, "Graphical models via generalized linear models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1358–1366. 27

[142] E. Yang, Y. Baker, P. Ravikumar, G. Allen, and Z. Liu, "Mixed graphical models via exponential families," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2014, pp. 1042–1050. 5, 6, 20

[143] E. Yang, P. K. Ravikumar, G. I. Allen, and Z. Liu, "On poisson graphical models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1718–1726. 4

[144] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* ACM, 1999, pp. 42–49. 99

[145] X. You, Q. Li, D. Tao, W. Ou, and M. Gong, "Local metric learning for exemplar-based object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1265–1276, 2014. 14

[146] K. Yu, S. Yu, and V. Tresp, "Multi-label informed latent semantic indexing," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.* ACM, 2005, pp. 258–265. 85

[147] S. Yu, K. Yu, V. Tresp, and H.-P. Kriegel, "Multi-output regularized feature projection," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 12, pp. 1600–1613, 2006. 49

[148] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," *Biometrika*, pp. 19–35, 2007. 3

[149] B. Zhang, Y. Wang, and F. Chen, "Multilabel image classification via high-order label correlation driven active learning," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1430–1441, 2014. 14

[150] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, 2015. 85

[151] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.* ACM, 2010, pp. 999–1008. 83

[152] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007. 47, 49, 74, 85, 97

[153] ——, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, 2014. 13, 47, 84

[154] Y. Zhang and J. G. Schneider, "Multi-label output codes using canonical correlation analysis." in *Proc. Int. Conf. Artif. Intell. Stat.*, 2011, pp. 873–882. 85

[155] ——, "Maximum margin output coding," in *Proc. Int. Conf. Mach. Learn.* ACM, 2012, pp. 1575–1582. 50, 74, 85, 97

[156] T. Zhou, D. Tao, and X. Wu, "Compressed labeling on distilled labelsets for multi-label learning," *Mach. Learn.*, vol. 88, no. 1-2, pp. 69–126, 2012. 49, 85

[157] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, “Missing value estimation for mixed-attribute data sets,” *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, 2011. 7, 9, 19

[158] X. Zhu and X. Wu, “Class noise vs. attribute noise: A quantitative study,” *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004. 6

[159] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. 53