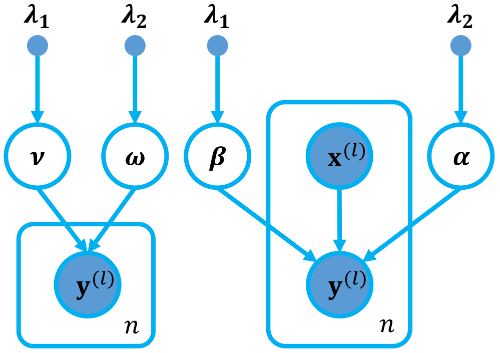
Conditional Graphical Lasso for Multi-label Image Classification

Qiang Li[1,2], Maoying Qiao[1], Wei Bian[1], Dacheng Tao[1]
[1]QCIS and FEIT, University of Technology Sydney
[2]Department of Computing, The Hong Kong Polytechnic University
{leetsiang.cloud, qiao.maoying}@gmail.com, {wei.bian, dacheng.tao}@uts.edu.au

Abstract

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 paper, 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 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.

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 [2], image annotation [15] and scene classification [5], 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 [29], dense SIFT [4], VLAD [18], object bank [25], and deep CNN [22, 8]. 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 [20]. A classical approach is the ChowLiu Tree [11] 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) [26] 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 [6, 43, 35]. An extension to the ChowLiu Tree is designed in [6] 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 [43]. More recently, clique generating machine (CGM) [35] 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 paper, 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 [28, 31, 24]. [1] We term the proposed method as conditional graphical Lasso (CGL). See Figure 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.

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 [36, 45, 16]. 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) [33], as well as its ensemble and probabilistic variants [10], 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) [37], pruned problem transformation (PPT) [32], hierarchical binary relevance (HBR) [7] and hierarchy of multi-label classifiers (HOMER) [36].

Algorithm adaptation methods extend typical classifiers to multi-label situation. For example, multi-label K nearest neighbour (MLkNN) [44] 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) [9] 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) [41], multi-label least square (MLLS) [19], multi-label F-statistics (MLF) and multi-label ReliefF (MLRF) [21]. Label specific features (LIFT) [42] 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 compressed sensing [17], random projection [47], principal label space transformation (PLST) [34] and maximum margin output coding (MMOC) [46].

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.

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 [31] 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

$$p(\mathbf{y}, \boldsymbol{\nu}, \boldsymbol{\omega}) = p(\mathbf{y}|\boldsymbol{\nu}, \boldsymbol{\omega}) p(\boldsymbol{\nu}) p(\boldsymbol{\omega}), \tag{1}$$

$$p(\mathbf{y}|\boldsymbol{\nu}, \boldsymbol{\omega}) \propto \exp\left\{\sum_{i=1}^{m} \nu_i \mathbf{y}_i + \sum_{i<j} \omega_{ij} \mathbf{y}_i \mathbf{y}_j\right\}, \tag{2}$$

$$p(\boldsymbol{\nu}) \propto \lambda_1^{d/2} \exp(-\lambda_1 \|\boldsymbol{\nu}\|_2^2), \tag{3}$$

$$p(\boldsymbol{\omega}) \propto \lambda_2^{d/2} \exp(-\lambda_2 \|\boldsymbol{\omega}\|_1), \tag{4}$$

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

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

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.

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 $\mathbf{x}$,

$$p(\mathbf{y}, \Theta|\mathbf{x}) = p(\mathbf{y}|\mathbf{x}, \Theta)p(\Theta). \tag{5}$$

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 $\mathcal{G}$,

$$p(\mathbf{y}|\mathbf{x}, \Theta) \propto \exp\left\{\sum_{i=1}^{m} \nu_i(\mathbf{x})\mathbf{y}_i + \sum_{i<j} \omega_{ij}(\mathbf{x})\mathbf{y}_i\mathbf{y}_j\right\}. \tag{6}$$

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

$$\begin{cases} \nu_i(\mathbf{x}) = \beta_i^T\mathbf{x}, & \text{for } i \in \mathcal{V}; \\ \omega_{ij}(\mathbf{x}) = \alpha_{ij}^T\mathbf{x}, & \text{for } (i,j) \in \mathcal{E}. \end{cases} \tag{7}$$

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 $\mathbf{x}$, (6) is exactly an Ising model for $\mathbf{y}$. It can also be treated as a special instantiation of CRF [23], 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 [30] 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,

$$p(\Theta) = p(\boldsymbol{\beta})p(\boldsymbol{\alpha}) = \prod_{i=1}^{m} p(\beta_i) \prod_{i<j} p(\alpha_{ij}), \tag{8}$$

$$p(\beta_i) \propto \lambda_1^{d/2} \exp(-\lambda_1 \|\beta_i\|_2^2), \tag{9}$$

$$p(\alpha_{ij}) \propto \lambda_2^{d/2} \exp(-\lambda_2 \|\alpha_{ij}\|_2), \tag{10}$$

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.

4. Algorithms

In this section, we derive both inference and learning algorithms for CGL. Generally, the label space $\mathcal{Y} = \{-1, 1\}^m$ in (6) maintains an exponentially large number of possible configurations. To normalize the conditional distribution in (6), one requires the log-partition function. For CGL with linear weight functions of $\mathbf{x}$ in (7), the log-partition function is defined as

$$A(\Theta, \mathbf{x}) = \log \sum_{\mathbf{y} \in \mathcal{Y}} \exp\left\{\sum_{i=1}^{m} \mathbf{y}_i\beta_i^T\mathbf{x} + \sum_{i<j} \mathbf{y}_i\mathbf{y}_j\alpha_{ij}^T\mathbf{x}\right\}, \tag{11}$$

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.

4.1. 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

$$\widehat{q}(\mathbf{y}) = \arg \min_{\substack{q(\mathbf{y})= \\ \prod_i q(\mathbf{y}_i)}} \text{KL}[q(\mathbf{y}) \| p(\mathbf{y}|\mathbf{x}, \Theta)]. \tag{12}$$

According to [3], the marginal $q(\mathbf{y}_i)$ that minimizes (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

```
Algorithm 1 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,
        $\xi_q(\mathbf{y}_{\setminus i}) = \left\{ \begin{array}{l} \mathbb{E}_{q^{(t+1)}(\mathbf{y}_j)}[\mathbf{y}_j] : 1 \leq j < i; \\ \mathbb{E}_{q^{(t)}(\mathbf{y}_j)}[\mathbf{y}_j] : i < j \leq m. \end{array} \right\}$
      Update the variational distribution $q^{(t+1)}(\mathbf{y}_i)$ with $\xi_q(\mathbf{y}_{\setminus i})$ by using (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
```

probability distribution. For brevity of presentation, we simply give the update formula for each $q(\mathbf{y}_i)$,

$$q(\mathbf{y}_i) \leftarrow \frac{1}{Z_i} \exp \mathbb{E}_{q(\mathbf{y}_{\setminus i})}[\ln p(\mathbf{y}|\mathbf{x}, \Theta)], \tag{13}$$

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 (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

$$\begin{aligned} & \mathbb{E}_{q(\mathbf{y}_{\setminus i})}[\ln p(\mathbf{y}|\mathbf{x}, \Theta)] \\ & = \mathbf{y}_i \beta_i^T \mathbf{x} + \mathbf{y}_i \mathbb{E}_{q(\mathbf{y}_{\setminus i})} \left[\sum_{j \neq i} \mathbf{y}_j \right] \alpha_{ij}^T \mathbf{x} + \text{const} && (14) \\ & = \mathbf{y}_i \beta_i^T \mathbf{x} + \mathbf{y}_i \sum_{j \neq i} \mathbb{E}_{q(\mathbf{y}_j)}[\mathbf{y}_j] \alpha_{ij}^T \mathbf{x} + \text{const}, && (15) \end{aligned}$$

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

$$q(\mathbf{y}_i) = \sigma \left(2\mathbf{y}_i \left(\beta_i^T \mathbf{x} + \sum_{j \neq i} \mathbb{E}_{q(\mathbf{y}_j)}[\mathbf{y}_j] \alpha_{ij}^T \mathbf{x} \right) \right), \tag{16}$$

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 1 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.

```
Algorithm 2 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 $\nabla J_s(\Theta)$ (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 1.
    Calculate the gradient of $J_s(\Theta)$ at $\Theta^{(k)} = (\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)})$ according to (24).
    Update $\Theta^{(k+1)} = (\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)})$ by using (27);
      $k = k + 1$
  end while
```

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 $q(\mathbf{y})$.

4.2. 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

$$\begin{aligned} \widehat{\Theta} & = \arg\max_{\Theta} p(\Theta|\mathbf{Y}, \mathbf{X}) && (17) \\ & = \arg\max_{\Theta} \frac{p(\mathbf{Y}, \Theta|\mathbf{X})}{\int_\Theta p(\mathbf{Y}, \Theta|\mathbf{X})} && (18) \\ & = \arg\max_{\Theta} p(\mathbf{Y}, \Theta|\mathbf{X}) && (19) \\ & = \arg\max_{\Theta} \prod_{l=1}^n p(\mathbf{y}^{(l)}|\mathbf{x}^{(l)}, \Theta) p(\Theta). && (20) \end{aligned}$$

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 (20) is achieved by considering (5) and the i.i.d. assumption.

By taking negative logarithm of the posterior and substituting (6), (9) and (10) into (20), the original maximization problem can be reformulated into an equivalent minimization problem as below,

$$
\begin{aligned}
\widehat{\Theta} = \arg\min_{\Theta} & -\sum_{i=1}^{m} \beta_i^T \bar{\phi}_i - \sum_{i<j} \alpha_{ij}^T \bar{\psi}_{ij} + \frac{1}{n}\sum_{l=1}^{n} A(\Theta, \mathbf{x}^{(l)}) \\
& + \frac{\lambda_1}{n}\sum_{i=1}^{m} \|\beta_i\|_2^2 + \frac{\lambda_2}{n}\sum_{i<j} \|\alpha_{ij}\|_2, \qquad (21)
\end{aligned}
$$

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 (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 (21). To learn the parameters Θ, a direct gradient-based optimizer is inapplicable due to the nonsmooth $\ell_{2,1}$-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.,

$$
\begin{aligned}
J_s(\Theta) = & -\sum_{i=1}^{m} \beta_i^T \bar{\phi}_i - \sum_{i<j} \alpha_{ij}^T \bar{\psi}_{ij} + \frac{1}{n}\sum_{l=1}^{n} A(\Theta, \mathbf{x}^{(l)}) \\
& + \frac{\lambda_1}{n}\sum_{i=1}^{m} \|\beta_i\|_2^2. \qquad (22)
\end{aligned}
$$

Further, according to the mean field approximation described in Section 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

$$
\left\{ \begin{array}{l}
\nabla A_{\beta_i}(\Theta, \mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\mathbf{y}_i\mathbf{x}] \approx \mathbb{E}_{\widehat{q}(\mathbf{y})}[\mathbf{y}_i\mathbf{x}] \\
\nabla A_{\alpha_{ij}}(\Theta, \mathbf{x}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{x})}[\mathbf{y}_i\mathbf{y}_j\mathbf{x}] \approx \mathbb{E}_{\widehat{q}(\mathbf{y})}[\mathbf{y}_i\mathbf{y}_j\mathbf{x}].
\end{array} \right. \qquad (23)
$$

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

$$
\left\{ \begin{array}{l}
\nabla J_{s\beta_i}(\Theta^{(k)}) \approx -\bar{\phi}_i + \frac{1}{n}\sum\limits_{l=1}^{n} \widehat{q}(\mathbf{y}_i^{(l)})\mathbf{x}^{(l)} + \frac{2\lambda_1}{n}\beta_i^{(k)} \\
\nabla J_{s\alpha_{ij}}(\Theta^{(k)}) \approx -\bar{\psi}_{ij} + \frac{1}{n}\sum\limits_{l=1}^{n} \widehat{q}(\mathbf{y}_i^{(l)})\widehat{q}(\mathbf{y}_j^{(l)})\mathbf{x}^{(l)}.
\end{array} \right. \qquad (24)
$$

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

$$
\begin{aligned}
J(\Theta; \Theta^{(k)}) = & \, J_s(\Theta^{(k)}) \\
& + \sum_{i=1}^{m} \langle \nabla J_{s\beta_i}(\Theta^{(k)}), \beta_i - \beta_i^{(k)} \rangle + \frac{1}{2\eta}\|\beta_i - \beta_i^{(k)}\|_2^2 \\
& + \sum_{i<j} \langle \nabla J_{s\alpha_{ij}}(\Theta^{(k)}), \alpha_{ij} - \alpha_{ij}^{(k)} \rangle \\
& + \frac{1}{2\eta}\|\alpha_{ij} - \alpha_{ij}^{(k)}\|_2^2 + \frac{\lambda_2}{n}\|\alpha_{ij}\|_2. \qquad (25)
\end{aligned}
$$

The parameter η in (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 (25), i.e.,

$$
\Theta^{(k+1)} = \arg\min_{\Theta} J(\Theta; \Theta^{(k)}), \qquad (26)
$$

which is solved by

$$
\left\{ \begin{array}{l}
\beta_i^{(k+1)} = \beta_i^{(k)} - \eta\nabla J_{s\beta_i}(\Theta^{(k)}) \\
\alpha_{ij}^{(k+1)} = \mathcal{S}(\alpha_{ij}^{(k)} - \eta\nabla J_{s\alpha_{ij}}(\Theta^{(k)}); \frac{\lambda_2}{n}),
\end{array} \right. \qquad (27)
$$

where the soft thresholding function is

$$
\mathcal{S}(u; \rho) = \left\{ \begin{array}{ll}
(1 - \frac{\rho}{\|u\|_2})u, & \text{if } \|u\|_2 > \rho; \\
0, & \text{otherwise.}
\end{array} \right. \qquad (28)
$$

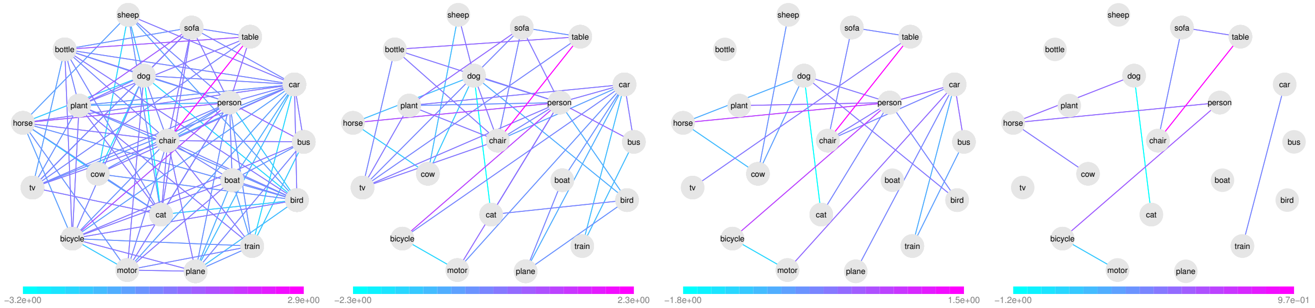
Iteratively applying (27) until convergence provides a first-order method for solving (21). The pseudo code for this procedure is summarized in Algorithm 2. Note that the gradient descent steps in Algorithm 2 can be speeded up with modern optimization procedures, such as the fast iterative shrinkage thresholding [1].

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.

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)[2], PASCAL VOC 2007 (PASCAL07) [12] and PASCAL VOC 2012 (PASCAL12) [13]. MULAN scene dataset contains 2047 images with 6 labels, and each image is represented

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



(a) #Edges: 109.

(b) #Edges: 46.

(c) #Edges: 25.

(d) #Edges: 10.

Figure 2. Illustration of the CGL label graphs learned from PASCAL07-CNN.

Table 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

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 [4] and the deep CNN (convolutional neural network) features [22, 8]. We extract PHOW features of 3600 dimensions by using the VLFeat implementation [38]. For deep CNN features, we use MatConvNet matlab toolbox [39] and the 'imagenet-vgg-f' model pretrained on ImageNet database [8] to represent each image as a 4096-dimensional feature. The basic information of the datasets is summarized in Table 1.

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 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.

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. In this paper, 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) [9], multi-label k-nearest neighbor (MLkNN) [44], classifier chains (CC) [33], maximum margin output coding (MMOC) [46], probabilistic label enhancement model (PLEM) [26] and clique generating machine (CGM) [35] 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 [14] 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 [3] and the authors' homepages [4] [5].

We use six widely accepted performance criteria to e-

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

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

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

Table 2. Multi-label image classification performance comparison via 5-fold cross validation

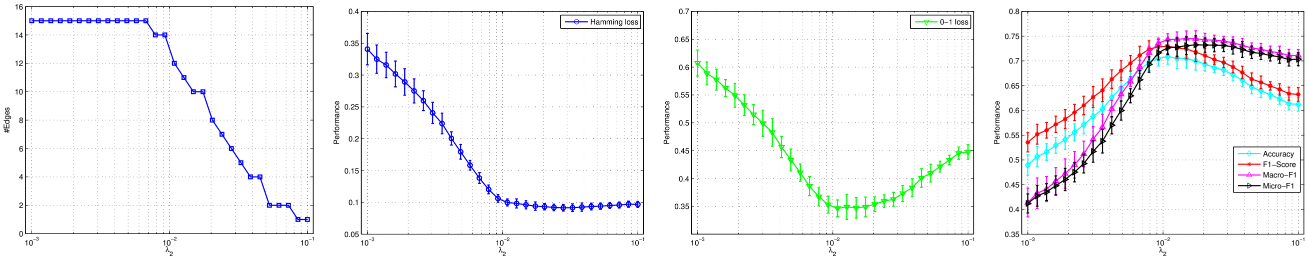
Datasets	Methods	Measures					
		Hamming loss	0-1 loss	Accuracy	F1-Score	Macro-F1	Micro-F1
MULANscene	ILRs	0.117±0.006	0.495±0.022	0.592±0.016	0.622±0.014	0.677±0.016	0.669±0.014
	IBLR	**0.085±0.004**	0.358±0.016	0.677±0.018	0.689±0.019	**0.747±0.010**	**0.738±0.014**
	MLkNN	0.086±0.003	0.374±0.015	0.668±0.018	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
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**
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**

valuate 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 [40, 27]. 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. For CGL hyperparameters, we fix $\lambda_1 = 0.01$, and set λ_2 as 0.0149 for MULANscene and 0.003 for PASCAL07 and PASCAL12.

5.3. Results and Discussion

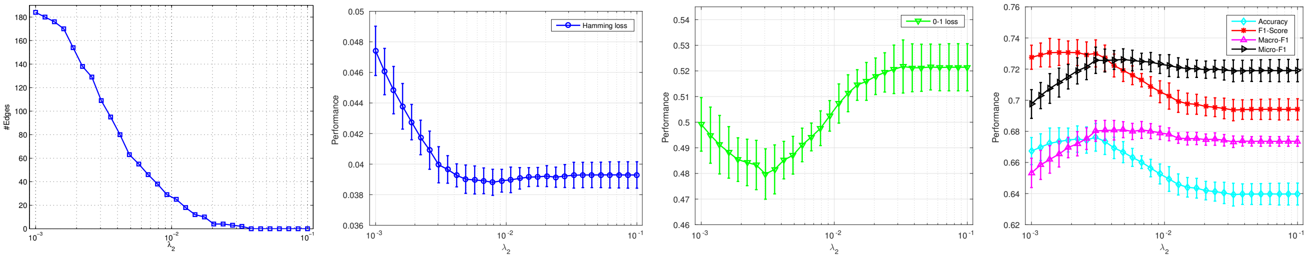
Table 2 summarizes the experimental results on MULANscene, PASCAL07 and PASCAL12 of all eight algorithms evaluated by the six measures. Except for Hamming loss, CGL achieves better or comparable results on all datasets with different types of feature. This is because Hamming loss treats the prediction 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



(a) #Edges (b) Hamming loss (c) 0-1 loss (d) Other measures

Figure 3. 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. Performance variation of CGL versus the hyperparameter λ_2 on PASCAL07-CNN.

methods (ILRs and CC). We observe that both CGL and C-C 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 5.1 by letting $\lambda_1 = 0.01$ and λ_2 range from 0.001 to 0.1. The results are shown in Figures 3 and 4. To make the performance variation easier to understand, we also provide the curve of #Edges versus λ_2 in Figures 3(a) and 4(a). 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.

6. Conclusion

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.

Acknowledgements This research is supported by Australian Research Council Projects DP-140102164, FT-130101457, and LE140100061.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[3] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[4] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1–8. IEEE, 2007.

[5] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognit.*, 37(9):1757–1771, 2004.

[6] J. K. Bradley and C. Guestrin. Learning tree conditional random fields. In *Proc. Int. Conf. Mach. Learn.*, pages 127–134, 2010.

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

[8] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[9] W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn.*, 76(2-3):211–225, 2009.

[10] W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proc. Int. Conf. Mach. Learn.*, pages 279–286, 2010.

[11] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inform. Theory*, 14(3):462–467, 1968.

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 (voc2012) results, 2012.

[14] 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.*, 9:1871–1874, 2008.

[15] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, volume 2, pages II–1002. IEEE, 2004.

[16] E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Comput. Surv.*, 47(3):52, 2015.

[17] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 22, pages 772–780, 2009.

[18] 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.*, pages 3304–3311. IEEE, 2010.

[19] S. Ji, L. Tang, S. Yu, and J. Ye. A shared-subspace learning framework for multi-label classification. *ACM Trans. Knowl. Discov. Data*, 4(2):8, 2010.

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

[21] 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.*, pages 2352–2359. IEEE, 2012.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1097–1105, 2012.

[23] 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.*, pages 282–289. ACM, 2001.

[24] J. Lee and T. Hastie. Structure learning of mixed graphical models. In *Proc. Int. Conf. Artif. Intell. Stat.*, pages 388–396, 2013.

[25] 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.*, pages 1378–1386, 2010.

[26] X. Li, F. Zhao, and Y. Guo. Multi-label image classification with a probabilistic label enhancement model. *Proc. Conf. Uncertain. Artif. Intell.*, 2014.

[27] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognit.*, 45(9):3084–3104, 2012.

[28] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, pages 1436–1462, 2006.

[29] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.*, 42(3):145–175, 2001.

[30] 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.*, pages 881–888. ACM, 2009.

[31] P. Ravikumar, M. J. Wainwright, J. D. Lafferty, et al. High-dimensional ising model selection using l1-regularized logistic regression. *Annals of Statistics*, 38(3):1287–1319, 2010.

[32] J. Read. A Pruned Problem Transformation Method for Multi-label classification. In *Proc. New Zealand Computer Science Research Student Conference*, pages 143–150, 2008.

[33] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Mach. Learn.*, 85(3):333–359, 2011.

[34] F. Tai and H.-T. Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

[35] 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.*, pages 4100–4109, 2015.

[36] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2010.

[37] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proc. Eur. Conf. Mach. Learn.*, pages 406–417. Springer, 2007.

[38] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[39] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proc. ACM Int. Conf. Multimedia*.

[40] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pages 42–49. ACM, 1999.

[41] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pages 258–265. ACM, 2005.

[42] M.-L. Zhang and L. Wu. Lift: Multi-label learning with label-specific features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1):107–120, 2015.

[43] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.*, pages 999–1008. ACM, 2010.

[44] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognit.*, 40(7):2038–2048, 2007.

[45] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837, 2014.

[46] Y. Zhang and J. G. Schneider. Maximum margin output coding. In *Proc. Int. Conf. Mach. Learn.*, pages 1575–1582. ACM, 2012.

[47] T. Zhou, D. Tao, and X. Wu. Compressed labeling on distilled labelsets for multi-label learning. *Mach. Learn.*, 88(1-2):69–126, 2012.