Multi-label Boosting for Image Annotation by Structural Grouping Sparsity

Fei Wu, Yahong Han
College of Computer Science
Zhejiang University, China
{wufei,yahong}@zju.edu.cn

Qi Tian
Department of Computer Science
University of Texas at San Antonio
qitian@cs.utsa.edu

Yueting Zhuang
College of Computer Science
Zhejiang University, China
yhuang@zju.edu.cn

ABSTRACT
We can obtain high-dimensional heterogeneous features from real-world images to describe their various aspects of visual characteristics, such as color, texture and shape etc. Different kinds of heterogeneous features have different intrinsic discriminative power for image understanding. The selection of groups of discriminative features for certain semantics is hence crucial to make the image understanding more interpretable. This paper formulates the multi-label image annotation as a regression model with a regularized penalty. We call it Multi-label Boosting by the selection of discriminative features for certain semantic tags. This paper formulates the multi-label selection of groups of discriminative features for certain semantic tags. Moreover, the correlations among multiple tags are utilized in MtBGS to boost the performance of multi-label annotation. Extensive experiments on public image datasets show that the proposed approach has better multi-label image annotation performance and leads to a quite interpretable model for image understanding.

Categories and Subject Descriptors
H.3 [INFORMATION STORAGE AND RETRIEVAL]: Content Analysis and Indexing; I.4.10 [IMAGE PROCESSING AND COMPUTER VISION]: Image Representation—Statistical

General Terms
Algorithms, Experimentation, Theory

Keywords
Multi-label Annotation, Structural Grouping Sparsity, Label Correlation

1. INTRODUCTION
Automatic annotating images with suitable multiple tags is a very active research field. From a machine learning point of view, the approaches of multi-label image annotation can be roughly classified into the generative model and the discriminative model. The generative model learns a joint distribution over image features and annotation tags. To annotate a new image, the learned generative model computes the conditional probability over tags given the visual features [1][7]. On the other hand, the discriminative model trains a separate classifier from visual features for each tag. These classifiers are used to predict particular tags for test image samples [12][4].

In real world images, we can extract high dimensional heterogeneous features from one given image, such as global features (color, shape and texture) or local features(SBN[19], SIFT, Shape Context and GLOH (Gradient Location and Orientation Histogram)). Different subsets of heterogeneous features have different intrinsic discriminative power to characterize one image label. That is to say, only limited groups of heterogeneous features distinguish each label from others. Therefore, the selected visual features to be used for the prediction of certain image semantics are usually sparse.

As a result, some sparsity-based multi-label learning approaches have been studied to impose structural penalty on feature and label spaces. For example, a multi-label sparse coding framework MLE[24] was proposed to utilize multi-label information for dimensionality reduction of visual features. Multi-task Sparse Discriminate Analysis (MtSDA) was implemented in [13] to avoid overfitting for highly correlated features and identify grouping effect in feature selection during multi-label annotation by $\ell_1$-norm and penalized matrix. Moreover, Cao [3] proposed to learn different metric kernel functions for different features. They formulated the Heterogeneous Feature Machines (HFM) as a sparse logistic regression by the $\ell_1$-norm at group level. For above approaches, the $\ell_1$-norm (namely lasso, least absolute shrinkage and selection operator) [23] was effectively implemented to make the learning model both sparse and interpretable. However, for group of features that the pairwise correlations among them are very high, lasso tends to select only one of the pairwise correlated features and does not induce the group effect. In the “large $p$, small $n$” problem, the “grouped features” situation is an important concern to facilitate a model’s interpretability. In order to remedy the deficiency of lasso, group lasso [25] and elastic net [29] were proposed respectively in the past years.
Figure 1: Flowchart of the heterogeneous feature selection with structural grouping sparsity.

This paper is interested in seeking after an interpretable model for predicting particular tags for images. Our goal is to select finite groups of heterogeneous features and identify subgroup within homogenous features. For example, as shown in Figure 1, lots of heterogeneous features such as color, texture and shape can be extracted from images. We tend to discern those discriminative feature sets from each image and set their selection coefficients ($\beta_i$) as 1 and the selection coefficients of other insignificant feature sets as 0. We then identify the subgroup within each selected feature set as the representation of each image.

Furthermore, in the setting of images with multiple labels, the effective utilization of the latent information hidden in related labels can boost the performance of multi-label annotation. Kang [17] proposed a Correlated Label Propagation (CLP) model through an efficiently solved submodular function. The CLP method utilized interactions between labels and simultaneously co-propagated multiple labels to function. The CLP method utilized interactions between labels and simultaneously co-propagated multiple labels to function. The CLP method utilized interactions between labels and simultaneously co-propagated multiple labels to function.

In order to achieve the goal of structural sparse feature selection and label correlation learning for multi-label image annotation, this paper proposes a framework of Multi-label Boosting by the selection of heterogeneous features with structural Grouping Sparsity (MtBGS).

MtBGS formulates the multi-label image annotation as a multiple response regression model with structural grouping penalty. A benefit of performing multi-label image annotation via regression is the ability to introduce penalties. Many of penalties can be introduced into the regression model for better prediction. Hastie [14] proposed the Penalized Discriminant Analysis (PDA) to tackle problems of overfitting in situations of large numbers of highly correlated predictors (features). PDA introduced a quadratic penalty with a symmetric and positive definite matrix $\Omega$ into the objective function. Elastic net [29] was proposed to conduct automatic variable selection and group selection of correlated variables simultaneously by imposing both $\ell_1$ and $\ell_2$-norm penalties. Furthermore, motivated by elastic net, Clemmensen [6] extended PDA to sparse discriminant analysis (SDA).

In this paper, we formulate MtBGS as a multi-response least square regression with structural grouping penalty. The basic motivation of imposing structural grouping penalty in MtBGS is to perform heterogeneous feature group selection and subgroup identification within homogeneous features simultaneously. As we know, some subgroups of features in high-dimensional heterogeneous features have more discriminative power for predicting certain labels of a given image. Furthermore, the correlations between labels are utilized by a Curds and Whey procedure [2] to boost the performance of image annotation in multi-label setting.

In particular, for group selection, we employ a (not squared) $\ell_2$-norm on the group-level coefficient vector. While for within group sparsity, the $\ell_1$-norm of the within group coefficient vector is imposed. Both the group-level $\ell_2$-norm and $\ell_1$-norm penalties are integrated to form the structural grouping penalty. Similar as sparse group lasso (group and piecewise penalty) [10] and group pursuit (pairwise penalty) [21], the structural grouping sparsity in this paper not only selects the groups of heterogeneous features, but also discriminates the subgroups within homogeneous features, which is most responsible for outcomes of a label. As a whole, our primary objective is to achieve an interpretable and accurate model for multi-label prediction through a computationally efficient method.

Here we have to point out that the proposed sparsity-based feature selection is different from other approaches such as visual diversity modeling, in which mixture of image kernels were integrated to characterize the diverse visual similarity relationships between images [9].

The remainder of this paper is organized as follows. We first introduce the framework of multi-label boosting by structural grouping sparsity and its computational issues in Section 2 and 3 respectively. The experimental analysis and conclusion are given in Section 4 and Section 5.

2. MULTI-LABEL BOOSTING BY STRUCTURAL GROUPING SPARSITY

2.1 Notation

Assume that we have a training set of $n$ labeled images with $J$ labels (tags): $\{(x_i, y_i) \in \mathbb{R}^p \times \{0, 1\}^J: i = 1, 2, \ldots, n\}$, where $x_i = (x_{i1}, \ldots, x_{ip})^T \in \mathbb{R}^p$ represents the predictors (features) vector for the $i$th image, $p$ represents the dimensionality of features. And $y_i = (y_{i1}, \ldots, y_{ij})^T \in \{0, 1\}^J$ is the corresponding label vector. $y_{ij} = 1$ if the $j$th label is the $j$th label and $y_{ij} = 0$ otherwise. Unlike the traditional multi-class problem where each sample only belongs to a single category: $\sum_{j=1}^J y_{ij} = 1$, in multi-label setting, we relax the constraint to $\sum_{j=1}^J y_{ij} \geq 0$. Let $X = (x_1, \ldots, x_n)^T$ be the $n \times p$ training data matrix, and $Y = (y_1, \ldots, y_n)^T$ be the corresponding $n \times J$ label indicator matrix.

Suppose we can extract $p$ high-dimensional heterogenous
features from images, and these \( p \) features are divided into \( L \) disjoint groups of homogeneous features, with \( p_l \) the number of features in the \( l \)th group, i.e., \( \sum_l p_l = p \). For ease of notation, we use a matrix \( X_l \in \mathbb{R}^{n \times p_l} \) to represent the features of training data corresponding to the \( l \)th group, with corresponding coefficient vector \( \beta_j \in \mathbb{R}^{p_l} (l = 1, 2, \ldots, L) \) for the \( j \)th label. Let \( \hat{\beta}_j = (\beta_{j1}^T, \ldots, \beta_{jL}^T)^T \) be the entire coefficient vector for the \( j \)th label, we have:

\[
X_j \beta_j = \sum_{l=1}^L X_l \beta_{jl}. \tag{1}
\]

In the following, we assume that the label indicator matrix \( Y \) is centered and the feature matrix \( X \) is centered and standardized, namely \( \sum_{i=1}^n y_{ij} = 0 \), \( \sum_{i=1}^n x_{id} = 0 \), and \( \sum_{i=1}^n x_{id}^2 = 1 \), for \( j = 1, 2, \ldots, J \) and \( d = 1, 2, \ldots, p \). Our algorithm can be generalized naturally to the unstandardized case. Moreover, we let \( ||\beta_j||_2 \) and \( ||\beta_j||_1 \) denote the \( \ell_2 \)-norm and the \( \ell_1 \)-norm of vector \( \beta_j \).

### 2.2 Problem Formulation and Solution

In this section, we describe the framework of the proposed Multi-label Boosting by selecting heterogeneous features with Structural Grouping Sparsity (MtBGS).

For the \( j \)th label, we tend to train a regression model with a penalty term as follows to select discriminative features:

\[
\min_{\beta_j} \left( ||Y_{(\cdot,j)} - \sum_{l=1}^L X_l \beta_{jl}||_2^2 + \lambda P(\beta_j) \right), \tag{2}
\]

where \( Y_{(\cdot,j)} \in (0,1)^{n \times 1} \) is the \( j \)th column of indicator matrix \( Y \) and encodes the label information for the \( j \)th label. \( P(\beta_j) \) is the regularizer which can impose certain structural priors of input data. For example, the ridge regression uses the \( \ell_2 \)-norm to avoid overfitting and lasso produces sparsity on \( \beta_j \) by the \( \ell_1 \)-norm.

Basically, MtBGS comprises two steps, namely regression with structural grouping penalty and multi-label boosting by curds and whey.

#### 2.2.1 Step 1: Regression with Structural Grouping Penalty

Since there are high-dimensional heterogeneous features in images, it is very natural to perform feature selection at group level (inter heterogeneous feature sets) first and then identify subgroup within a homogeneous feature set. The motivation of structural grouping penalty in MtBGS is to set most of coefficients in vectors \( \beta_{jl} \) to zero and only keep in the model limited number of coefficients, whose corresponding groups of features are discriminative to the \( j \)th label. That is to say, only discriminative subgroups of homogeneous features are selected out.

For each label \( j \) and its corresponding indicator vector, the regression model of MtBGS is defined as follows:

\[
\min_{\beta_j} \sum_{j=1}^J \left( ||Y_{(\cdot,j)} - \sum_{l=1}^L X_l \beta_{jl}||_2^2 + \lambda_1 \sum_{l=1}^L ||\beta_{jl}||_2 + \lambda_2 ||\beta_j||_1 \right), \tag{3}
\]

where \( \lambda_1 \sum_{l=1}^L ||\beta_{jl}||_2 + \lambda_2 ||\beta_j||_1 \) is the regularizer \( P(\beta_j) \) in (2) and is called the structural grouping penalty.

Let \( \hat{\beta}_{jl} \) be the solution of formula (3), we can predict the probability \( \tilde{y}_u \) that unlabeled images \( X_u \) belong to the \( j \)th label as follows:

\[
\tilde{y}_u = X_u \hat{\beta}_{jl}. \tag{4}
\]

#### 2.2.2 Step 2: Multi-label Boosting by Curds and Whey

The usual procedure of performing individual regression of each label on the common set of features ignores the correlations between labels. We propose to take advantage of correlations between labels to improve predictive accuracy. We call this method the multi-label boosting by Curds and Whey (C\&W) [2].

Curds and Whey sets up the connection between multiple response regression and canonical correlation analysis. Therefore, the C\&W method can be used to boost the performance of multi-label prediction given the prediction results from the individual regression of each label, and hence it can be easily integrated into our MtBGS framework.

### 2.3 Regularized Regression with Structural Grouping Penalty

Unlike group lasso, our structural grouping penalty in (3) not only selects the groups of heterogeneous features, but also identifies the subgroup of homogeneous features within each selected group.

Note that when \( \lambda_1 = 0 \), the formula (3) reduces to the traditional lasso under the multi-label learning setting, and \( \lambda_2 = 0 \) for the group lasso [25].

Therefore the framework of MtBGS puts forth a more flexible mechanism to the selection of heterogeneous features for the image understanding with multiple labels.

#### 2.3.1 Group Selection

For each label \( j \), we discuss how to obtain the coefficient vector \( \beta_j \) in the consequent sections. If \( \beta_{jl} \neq \{0\}^p_l \), it means that the \( l \)th group of homogeneous features is selected for the \( j \)th label.

According to [10], the subgradient equations of first two terms in (3) are

\[
-X_l^T (Y_{(\cdot,j)} - \sum_{l} X_l \hat{\beta}_{jl}) + \lambda_1 s_{jl} = 0; l = 1, \ldots, L; j = 1, \ldots, J \tag{5}
\]

where \( s_{jl} = ||\beta_{jl}||_2 ||\beta_{jl}||_2 \) if \( \beta_{jl} \neq \{0\}^p_l \) and \( s_{jl} \) is a vector with \( ||s_{jl}||_2 \leq 1 \) otherwise. We now focus on the solution for one group and hold other coefficients fixed.

Let the solutions of (5) to be \( \hat{\beta}_{j1}, \hat{\beta}_{j2}, \ldots, \hat{\beta}_{jL} \). If

\[
||X_l^T (Y_{(\cdot,j)} - \sum_{k=1}^L X_k \hat{\beta}_{jk})|| < \lambda_1 \tag{6}
\]

then \( \hat{\beta}_{jl} \) is set to zero; otherwise it satisfies

\[
\hat{\beta}_{jl} = (X_l^T X_l + \lambda_1 I/||\beta_{jl}||_2^2)^{-1} X_l^T (Y_{(\cdot,j)} - \sum_{k=1}^L X_k \hat{\beta}_{jk}). \tag{7}
\]

This leads to an algorithm that cycles through the groups, which is a blockwise coordinate descent procedure [25, 10]. The criterion (3) is convex and separable so that the blockwise coordinate descent at group level and piecewise coordinate descent within group for individual features can be used for optimization.

We first focus on just one group \( l \) of label \( j \), and denote the corresponding \( p_l \)-dimensional homogeneous features
of \( j \)th label by \( X_l = (X_l^1, X_l^2, \ldots, X_l^p) \). Since our structural grouping penalty attends to identify the subgroup within each selected group, we assume the \( l \)th group is selected and the coefficients \( \beta_{jl} = \theta_{jl} = (\theta_1, \theta_2, \ldots, \theta_p) \). We let \( r_{jl} = Y_{(\cdot,j)} - \sum_{k \neq l} X_k^j \beta_{jk} \) denote the partial residual when the \( l \)th group is removed.

The subgradient equations of (3) with respect to \( \theta_m \) are

\[
-X_l^m \, \theta_{jl} - \sum_{m=1}^{p_l} X_m \, \theta_{m} \right) + \lambda_1 \, s_m + \lambda_2 \, t_m = 0, \quad m = 1, \ldots, p_l
\]

where \( s_m = \theta_m / ||\theta_m||_2 \) if \( \theta_m \neq 0 \) and \( s_m \) is a vector satisfying \( ||s_m||_2 \leq 1 \) otherwise, and \( t_m \in \text{sign}(\theta_m) \), that is \( t_m = \text{sign}(\theta_m) \) if \( \theta_m \neq 0 \) and \( t_m \in [-1, 1] \) if \( \theta_m = 0 \).

Letting \( a = X^\top_l \, r_{jl} \), then a necessary and sufficient condition for \( \theta_{jl} \) to be zero, which means the \( l \)th group is dropped out of the model, is that the system of equation

\[
a_m = \lambda_1 s_m + \lambda_2 t_m
\]

has a solution with \( ||s_m||_2 \leq 1 \) and \( t_m \in [-1, 1] \). We can determine this by minimizing

\[
J(t) = \frac{1}{2} \sum_{m=1}^{p_l} (a_m - \lambda_2 t_m)^2 = \sum_{m=1}^{p_l} s_m^2
\]

with respect to \( t_m \in [-1, 1] \) and then check if \( J(t) \leq 1 \), which means \( \theta_{jl} = 0 \) and therefore the \( l \)th group is dropped out of the model.

Now if \( J(t) > 1 \), which means the \( l \)th group is selected, then we must minimize the following criterion to identify the subgroup of homogeneous feature sets after the \( l \)th group is selected:

\[
||r_{jl} - \sum_{m=1}^{p_l} X_m^l \theta_m||_2^2 + \lambda_1 ||\theta_{jl}||_2 + \lambda_2 \sum_{m=1}^{p_l} ||\theta_m||_1.
\]

Formula (11) is the sum of a convex differentiable function (first two terms) and a separable penalty. In next section we develop the Gauss-Seidel Coordinate Descent (GSCD) algorithm [22] to minimize (11) by a one-dimensional search over \( \theta_m \).

We summarize the algorithm for solving the regularized regression with structural grouping penalty in Algorithm 1.

### 2.3.2 Subgroup Identification by GSCD

We now derive the algorithm for solving the step 6 in Algorithm 1 to identify subgroup. We know that formula (11) is a convex function, therefore a global optimized result can be calculated.

The subgradient equation of (11) is

\[
g_m = -X_m^l \, r_{jl} - \sum_{m=1}^{p_l} X_m^l \theta_m + \lambda_1 \, s_m + \lambda_2 \, t_m = 0.
\]

Let us define

\[
\text{viol}_m = \begin{cases} 
|\lambda_2 - G_m| & \text{if } \theta_m > 0, \\
|\lambda_2 + G_m| & \text{if } \theta_m < 0, \\
\max(G_m - \lambda_2, -\lambda_2 - G_m, 0) & \text{if } \theta_m = 0,
\end{cases}
\]

where

\[
G_m = X_m^l \, r_{jl} + \sum_{m=1}^{p_l} X_m^l \theta_m - \lambda_1 \, s_m - \lambda_2 \, t_m.
\]

according to the Karush-Kuhn-Tucker (KKT) conditions, the first order optimality conditions for (11) can be written as

\[
\text{viol}_m \leq \tau, \quad \forall m \in \{1, 2, \ldots, p_l\}
\]

where \( \tau > 0 \) is the error tolerance. In fact, we refer to (15) as optimality with tolerance \( \tau \).

In solving (11) by Gauss-Seidel method, one variable \( \theta_m \) with the maximum \( \text{viol}_m \), violates the optimality conditions (15) is chosen and the optimization subproblem is solved with respect to this variable \( \theta_m \), alone, keeping the other \( \theta_m \) fixed. This procedure is repeated as long as there exists a variable which violates conditions (15).

Let us define the following two sets \( L = \{m : \theta_m = 0\} \) and \( I_{\theta_m} = \{m : \theta_m \neq 0\} \). The key to efficiently solve (11) by Gauss-Seidel methods is the selection of the variable \( \theta_m \) in each iteration with respect to which the objective function is optimized. A combination of the bisection method and Newton method [22] is used to optimize (11). In this method, two points \( L \) and \( H \) for which the derivative of the objective function in (11) has opposite signs are chosen, which ensures the root always lies in an interval \([L, H]\).

The minimizer computation through a Newton update is

\[
\theta_{m_{\text{new}}} = \theta_m - g_m / H_{mm},
\]

where \( H_{mm} \) is the diagonal elements of the Hessian \( H = \sum_{m=1}^{p_l} X_m^l \, X_m^l \) for (11) with respect to \( \theta_m \). This procedure can be best explained using Algorithm 2.

It is important to note that the objective function in (11) has different right-hand and left-hand derivatives with respect to \( \theta_m \) at \( \theta_m = 0 \). Therefore, in case when the current value of \( \theta_m \) is 0, we have to try both directions and compute \( g_m \) in the step 10 of Algorithm 2 according to the method in [11].

### 2.4 Multi-label Boosting by Curds and Whey

In order to take advantage of correlations between the labels to boost multi-label annotation, we propose to utilize the curds and whey (C&W) [2] method and integrate it into our MtBGS framework.

Let \( \hat{\beta}_j \) be the estimated coefficient vector for the \( j \)th label output by Algorithm 1, and \( \hat{y}_j \) denote corresponding
estimated indicator vector of the \( j \)th label, we have
\[
\hat{y}_j = X\hat{\beta}_j.
\]

According to [2], if the labels are correlated we may be able to obtain a more accurate prediction \( \hat{y}_j \) by a linear combination \( \hat{y}_j = By_j \). The matrix \( B \in \mathbb{R}^{J \times J} \) takes the form
\[
B = C^{-1}WC,
\]
where \( C \) is the \( J \times J \) matrix whose rows are the (\( y \)) canonical coordinates output by canonical correlation analysis (CCA) [15] and \( W = \text{diag}(w_1, w_2, \ldots, w_J) \) is a diagonal matrix. In this way, the C&W procedure is a form of multivariate shrinking of \( \hat{y}_j \). It transforms (by \( C \)), shrinks (multiplies by \( W \)) and then transforms back (by \( C^{-1} \)). In an idealized setting that the i.i.d predicting errors are independent of the labels, the optimal shrinkage matrix \( B^* \) can be derived by CCA. CCA computes two canonical coordinates vectors, \( v_{xj} \in \mathbb{R}^p \) and \( v_{yj} \in \mathbb{R}^q \), such that the following correlation coefficient
\[
\rho_j = \frac{\langle v_{xj}^T X^T Y v_{yj} \rangle}{\sqrt{\langle v_{xj}^T X^T X v_{xj} \rangle \langle v_{yj}^T Y^T Y v_{yj} \rangle}}
\]
is maximized, where \( j = 1, 2, \ldots, J \) (supposes \( J < p \) here). Breiman and Friedman [2] derived that the rows of the matrix \( C \) in (18) are the (\( y \)) canonical coordinates \( v_{yj}^T \), and \( w_j (j = 1, 2, \ldots, J) \) in matrix \( W \) are
\[
w_j = \frac{\rho_j^2}{\rho_j^2 + \gamma(1 - \rho_j^2)},
\]
where \( \gamma = p/n \). We summarize the multi-label boosting by Curds and Whey method in Algorithm 3.

3. COMPUTATIONAL DISCUSSION

3.1 Complexity

The computational complexity is crucial for the successful application of an algorithm. In Tibshirani’s original paper [23], he has found that the model selection with \( \ell_1 \)-norm usually can be finished with the iteration number within the range of \( (0.5p, 0.75p) \) in practice. The run-time performance of regression model with structural grouping sparsity is also very efficient when implemented by the cyclic coordinate descent method. From the description of coordinate descent by Gauss-Seidel method, we can see that for a complete cycle through all the coordinates, it takes \( O(k) \) operations, where \( k \) is the number of non-zero elements, when sparsity of the data is considered. So the complexity of the regression model with structural grouping sparsity is roughly \( O(p \times n) \).

3.2 Convergence

Zhang and Oles [26] also used the coordinate descent optimization method to solve the ridge logistic regression. The convergence of the coordinate descent has been verified in [26]. The basic idea is that after each iteration, the value of the objective function decreases strictly. Because of the convexity of the objective function, the optimization will definitely converge to its global minimal. As discussed above, the objective function in formula (11) is convex. Therefore, the solutions of coefficient vector \( \hat{\beta}_j \) of structural grouping sparsity for each label \( j \) are guaranteed to converge to the global optimality. Since at each step (each round of the repeat iteration) of Algorithm 2, the Karush-Kuhn-Tucker (KKT) condition does not hold, and the objective function in formula (11) decreases. Therefore, this guarantees the convergence of Algorithm 2 thanks to the convexity of the objective function in (11).

3.3 Stability

Stability here means that the performance of regression prediction cannot be affected by the (training) sample selection. As discussed in [2], since \( \hat{\beta}_j \) output by Algorithm 1 are estimated from the training data \( X \) and \( Y \), the original sample selection in first step of Algorithm 3 will overestimate the canonical correlations in CCA.

In order to evaluate the stability of the C&W method, we compare two configurations of the sample selection in the first step of Algorithm 3. The first one is the current setting in Algorithm 3. The second configuration is to randomly resample some samples from test dataset (different part from training data \( X \) and label indicators \( Y \)) and perform CCA for the first step of Algorithm 3. To investigate the stability we explore three different re-sampling methods, i.e., Cross Validation, Jackknife [20], and Bootstrap [8]. Details are reported in the experiments.

4. EXPERIMENTS

In this section, we systematically evaluate the effectiveness...
of our proposed MtBGS framework in automatic multi-tag image annotation.

4.1 Experimental Configuration

4.1.1 Dataset

Three benchmark image datasets are used in our experiments: Microsoft Research Cambridge (MSRC), MIML [28], and NUS-WIDE [5]. 23 and 5 class labels (tags) are respectively associated with images in MSRC and MIML, which are multi-tagged and can be used as annotation ground truth. We randomly sampled 10,000 images from NUS-WIDE in our experiments. For the ground truth of NUS-WIDE we chose two indicator matrices from the selected data samples to form two datasets - NUS-6 and NUS-16. In these two datasets, the top 6 and 16 tags which label the maximum numbers of positive instances were selected respectively.

Multiple heterogeneous features were extracted and concatenated as a visual feature vector for each image. Taking each type of homogeneous features as a group, we sequentially numbered the feature groups in the following sections. Details of features, dimensionality, and also the sequence numbers for each of the three datasets are listed as follows:

**MSRC** and **MIML**: 638-D features are sequentially divided into 7 groups. 1: 256-D color histogram; 2: 6-D color moments; 3: 128-D color coherence; 4: 15-D textures; 5: 10-D tamura-texture coarseness; 6: 8-D tamura-texture directionality; 7: 80-D edge orientation histogram; 8 (only for MIML): 135-D SBN colors [19]. Note that the 8th group of features is only used in MIML dataset. Therefore, the dimensionality of visual feature vector for images in MSRC is 503-D.

**NUS**: 634-D features are sequentially divided into 5 groups. 1: 64-D color histogram; 2: 144-D color correlogram; 3: 73-D edge direction histogram; 4: 128-D wavelet texture; 5: 225-D block-wise color moments.

4.1.2 Evaluation Metrics

The area under the ROC curve (AUC) and F1 score are used to measure the performance of each tag. Since there are multiple labels (tags) in our experiments, to measure both the global performance across multiple tags and the average performance of all tags, according to [18][16] we use both the microaveraging and macroaveraging methods.

4.2 Parameter Tuning

The parameters $\lambda_1$ and $\lambda_2$ in (3) need to be tuned. At the first training/test partition we choose those parameters by a 5-fold cross validation on the training dataset. These chosen parameters were then fixed to train the MtBGS model for the 10 partitions. Note that, different features play different roles in our MtBGS framework for different labels. Therefore, the parameter tuning process is performing separately for each tag. We depict 3 examples of parameter tuning by the 5-fold cross validation with respect to micro-AUC and macro-AUC.
Figure 5: Heterogeneous feature selection results from group lasso, lasso, and MtBGS for 10-round repetition of label “tree” in MSRC dataset. Different colors indicate different rounds.

Figure 6: Heterogeneous feature selection results from group lasso, lasso, and MtBGS for 10-round repetition of label “bird” in MSRC dataset. Different colors indicate different rounds.
4.3 Heterogeneous Feature Selection

We explored the differences of heterogeneous feature selection between group lasso, lasso, and MtBGS on MSRC and MIML dataset for each label respectively. Sample images for 2 instance labels from MSRC and MIML are listed in Figure 3 and Figure 4, respectively. As can be seen, images with different labels (semantics) have different heterogeneous low-level features, such as color and texture etc. Therefore, training a model for heterogeneous feature selection is crucial for understanding the semantic content in these images.

In order to uncover the different mechanism of heterogeneous feature selection for group lasso, lasso and MtBGS, we output the coefficient vectors \( \beta \) respectively from the three algorithms after 10-round repetition, and investigate the results of group selection for each round. The results of heterogeneous feature selection for 2 labels in Figure 3 are depicted in Figure 5 and Figure 6. We observe that though group lasso and MtBGS can both select groups of features, the coefficient values of \( \beta \) within groups are obviously different. MtBGS successfully induces sparsity of coefficient values, i.e., shrink to zeros, within groups, which is like lasso. On the contrary, group lasso intends to include all the coefficients into the model. Comparing the group selection results between group lasso and MtBGS, MtBGS produces more consistent group selection with respect to the 10-round repetition training process. Moreover, considering the heterogeneous feature selection for the label “bird”, MtBGS adaptively includes more groups of heterogeneous features into the model. Since the low-level features in images with label “bird” are more variant.

Furthermore, we explore the results of heterogeneous feature selection by MtBGS for different size of training data. Let’s see Figure 7, we depict the results by MtBGS for images of of label “mountain” in MIML dataset with different size of training data. Note that the coefficient values are plotted from one round, and the group selection are plotted from 10-round repetition. As can be seen, the heterogeneous feature selection is more consistent and interpretable when the size of training data is increasing. For example, the most discriminative features for label “mountain” (see sample images in Figure 4) are color and shape. The corresponding extracted feature groups in our experiments are: 1: 256-D color histogram, 2: 6-D color moments, 3: 128-D color coherence, 8: 135-D SBN, and 7: 80-D edge orientation histogram. Comparing the results in Figure 7, noisy feature groups, i.e., the texture feature groups, are almost excluded from the models when the number of training data reaches 900. In particular, the coefficient vector output from the 900 training samples is more sparse.

4.4 Performance of Multi-label Learning and Annotation

We first investigate the learning performance of MtBGS before C&W. MtBGS for each label is trained separately on the training data with different size, i.e., \{500, 600, 700, 800, 900, 1,000\} and \{1,000, 1,2000, 1,400, 1,600, 1,800, 2,000\} for MIML and NUS-6 respectively. This process is repeated for 10-round by randomly sampling 10 times for each size of training data. From Figure 8 we can observe that the MtBGS produces better learning performance with more training samples. Moreover, the performance improving ratio of
The average performance and standard deviations from 10-round of repetition for each algorithm are reported in Table 2. We can see that the MtBGS framework outputs the best image annotation results in terms of AUC and F1 score. Furthermore, the performance of multi-label annotation by MtBGS from NUS-6 is better than from NUS-16, since the correlations between tags in NUS-6 are relative more dense than those in NUS-16.

5. CONCLUSION

This paper proposes a framework of multi-label learning for image annotation, called the MtBGS. The MtBGS method is attractive due to its subgroup feature identification by structural grouping penalty in heterogeneous fea-

---

Table 1: Multi-label boosting and stability by C&W on MIML and NUS-6 dataset.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Before C&amp;W</th>
<th>C&amp;W(training)</th>
<th>C&amp;W(test)</th>
<th>Cross validation</th>
<th>Jackknife</th>
<th>Bootstrap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MIML</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>micro-AUC</td>
<td>0.7329±0.0175</td>
<td>0.7332±0.0177</td>
<td>0.7423±0.0163</td>
<td>0.7425±0.0163</td>
<td>0.7425±0.0163</td>
<td>0.7391±0.0167</td>
</tr>
<tr>
<td>macro-AUC</td>
<td>0.7368±0.0178</td>
<td>0.7371±0.0180</td>
<td>0.7457±0.0164</td>
<td>0.7459±0.0164</td>
<td>0.7459±0.0164</td>
<td>0.7429±0.0168</td>
</tr>
<tr>
<td>micro-F1</td>
<td>0.5209±0.0162</td>
<td>0.5210±0.0158</td>
<td>0.5251±0.0142</td>
<td>0.5250±0.0145</td>
<td>0.5252±0.0142</td>
<td>0.5248±0.0145</td>
</tr>
<tr>
<td>macro-F1</td>
<td>0.5199±0.0169</td>
<td>0.5200±0.0169</td>
<td>0.5237±0.0150</td>
<td>0.5237±0.0153</td>
<td>0.5239±0.0150</td>
<td>0.5236±0.0153</td>
</tr>
<tr>
<td><strong>NUS-6</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>micro-AUC</td>
<td>0.7749±0.0090</td>
<td>0.7834±0.0094</td>
<td>0.7840±0.0091</td>
<td>0.7841±0.0090</td>
<td>0.7841±0.0091</td>
<td>0.7830±0.0091</td>
</tr>
<tr>
<td>macro-AUC</td>
<td>0.7680±0.0084</td>
<td>0.7756±0.0086</td>
<td>0.7763±0.0083</td>
<td>0.7764±0.0083</td>
<td>0.7765±0.0084</td>
<td>0.7753±0.0083</td>
</tr>
<tr>
<td>micro-F1</td>
<td>0.5029±0.0118</td>
<td>0.5077±0.0125</td>
<td>0.5127±0.0127</td>
<td>0.5128±0.0127</td>
<td>0.5127±0.0127</td>
<td>0.5119±0.0125</td>
</tr>
<tr>
<td>macro-F1</td>
<td>0.4894±0.0099</td>
<td>0.4952±0.0105</td>
<td>0.4979±0.0107</td>
<td>0.4980±0.0106</td>
<td>0.4981±0.0107</td>
<td>0.4971±0.0106</td>
</tr>
</tbody>
</table>

---

Table 2: Multi-label annotation comparison on NUS-6 and NUS-16 dataset.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>CCA-ridge</th>
<th>CCA-SVM</th>
<th>SVM</th>
<th>MTL-LS</th>
<th>MtBGS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NUS-6</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>micro-AUC</td>
<td>0.5153±0.0374</td>
<td>0.6019±0.0321</td>
<td>0.6074±0.0388</td>
<td>0.5856±0.0310</td>
<td>0.7840±0.0091</td>
</tr>
<tr>
<td>macro-AUC</td>
<td>0.5608±0.0053</td>
<td>0.5598±0.0087</td>
<td>0.5630±0.0107</td>
<td>0.5666±0.0057</td>
<td>0.7763±0.0083</td>
</tr>
<tr>
<td>micro-F1</td>
<td>0.2713±0.0360</td>
<td>0.2645±0.0580</td>
<td>0.2480±0.0415</td>
<td>0.2283±0.0194</td>
<td>0.5127±0.0127</td>
</tr>
<tr>
<td>macro-F1</td>
<td>0.1408±0.0208</td>
<td>0.1234±0.0291</td>
<td>0.1086±0.0346</td>
<td>0.2055±0.0056</td>
<td>0.4981±0.0107</td>
</tr>
<tr>
<td><strong>NUS-16</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>micro-AUC</td>
<td>0.6594±0.1199</td>
<td>0.6611±0.0929</td>
<td>0.6716±0.1013</td>
<td>0.6872±0.1248</td>
<td>0.7657±0.0085</td>
</tr>
<tr>
<td>macro-AUC</td>
<td>0.7062±0.0629</td>
<td>0.7061±0.0666</td>
<td>0.7107±0.0777</td>
<td>0.7308±0.0873</td>
<td>0.7336±0.0088</td>
</tr>
<tr>
<td>micro-F1</td>
<td>0.2008±0.0996</td>
<td>0.3255±0.0820</td>
<td>0.3363±0.0978</td>
<td>0.2224±0.1106</td>
<td>0.3883±0.0156</td>
</tr>
<tr>
<td>macro-F1</td>
<td>0.1801±0.0728</td>
<td>0.1723±0.0774</td>
<td>0.1738±0.0935</td>
<td>0.2077±0.0889</td>
<td>0.2919±0.0098</td>
</tr>
</tbody>
</table>

---

Figure 8: AUC and F1 score versus different size of training data for MIML and NUS-6 dataset by MtBGS.
tured settings along with its multi-label boosting capability. Experiments show that the MtBGS not only is more interpretable for image annotation, but also achieves better results than the CCA-ridge, CCA-SVM, SVM, and MTL-LS methods.

6. ACKNOWLEDGMENTS
This work is supported by NSF (No.90920303), 973 program (2010CB327905), 863 Program (2006AA010107) and Program for Changjiang Scholars and Innovative Research Team in University (IRT0652, PCSIRT). This work is supported in part by Akiira Media Systems, Inc. for Dr. Qi Tian.

7. REFERENCES


