This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

ML-MG: Multi-label Learning with Missing Labels Using a Mixed Graph

Baoyuan Wu
KAUST
Saudi Arabia
baoyuan.wu@kaust.edu.sa

Siwei Lyu
University at Albany, SUNY
Albany, NY, USA
slyu@albany.edu

Bernard Ghanem
KAUST
Saudi Arabia
bernard.ghanem@kaust.edu.sa

Abstract

This work focuses on the problem of multi-label learning with missing labels (MLML), which aims to label each test instance with multiple class labels given training instances that have an incomplete/partial set of these labels (i.e. some of their labels are missing). To handle missing labels, we propose a unified model of label dependencies by constructing a mixed graph, which jointly incorporates (i) instance-level similarity and class co-occurrence as undirected edges and (ii) semantic label hierarchy as directed edges. Unlike most MLML methods, We formulate this learning problem transductively as a convex *quadratic matrix optimization problem that encourages training label consistency and encodes both types of label dependencies (i.e. undirected and directed edges) using quadratic terms and hard linear constraints. The alternating direction method of multipliers (ADMM) can be used to exactly and efficiently solve this problem. To evaluate our proposed method, we consider two popular applications (image and video annotation), where the label hierarchy can be derived from Wordnet. Experimental results show that our method achieves a significant improvement over state-of-the-art methods in performance and robustness to missing labels.*

1. Introduction

In multi-label learning, we assume that an instance can be assigned to multiple classes simultaneously. For example, an image can be annotated using several tags, and a document can be associated with multiple topics. Although many multi-label learning methods have been proposed, many of such methods require completely labeled training instances. In practical applications, most training instances are only partially labeled, with some or all of the labels not provided/missing. Let us consider the task of large-scale image annotation, where the number of classes/tags is large (e.g. using labels of ImageNet [12]). In this case, a human annotator can only realistically annotate each training image with a subset of tags. Learning from such partially labeled instances is referred to as the *multi-label learning with missing labels* (MLML) problem [39, 45].

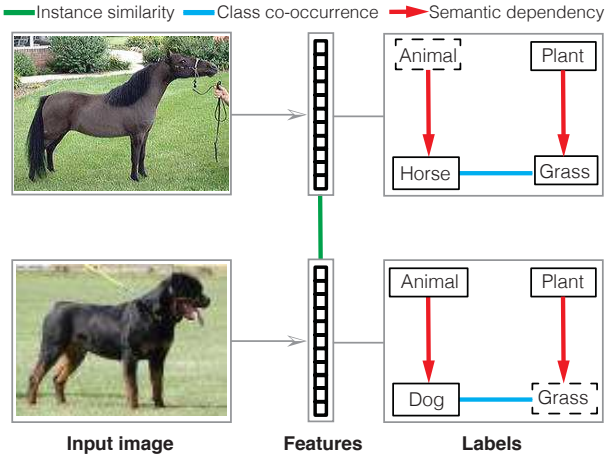


Figure 1: The left column includes two example images from the ESP Game [34] dataset, and their corresponding features and labels are shown in the right column. The solid box denotes a provided label, while the dashed box indicates a missing label. The red (semantic hierarchy), blue (class co-occurrence), and green (instance similarity) edges constitute a mixed graph. For clarity, we only present the relevant labels and edges, while other candidate labels and edges are ignored.

As labels are usually related by semantic meanings or co-occurrences, the key to filling and learning from missing labels is a good model to represent label dependency. One widely used model for label dependency is an undirected graph, through which the label information can be propagated among different instances and among different classes. For example, the label dependency between a pair of labels, such as instance similarity and class co-occurrence, can be represented using such a graph (see green and blue edges in Figure 1). However, as stated in [39, 41], class co-occurrence that is derived from training labels can be inaccurate and even detrimental when many missing labels exist. Li et al. [25] propose to alleviate this limitation by using an auxiliary source (such as Wikipedia)

to estimate co-occurrence relations.

Interestingly, the semantic dependency between two classes, such as "animal–>horse" and "plant–>grass" as shown in Figure 1, can foster further label dependency and improve label predictions in the test. In this case, it is intuitive to require that *the label score* (e.g. the presence probability) *of the parent class cannot be lower than that of its child class*. This is traditionally referred to as the **semantic hierarchy constraint** [2]. The undirected graph (with instance similarity and class co-occurrence edges) cannot guarantee that the final label predictions will satisfy all semantic hierarchy constraints. To address this problem, we add semantic dependencies into the graph as directed edges, thus, resulting in an overall mixed graph that encourages (or enforces) three types of label dependency (refer to Figure 1 for an example).

The goal of this work is to learn from partially labeled training instances and to correctly predict the labels of test instances, which should satisfy the semantic hierarchy constraints. Firstly, motivated by [39, 41], a discrete objective function is formulated to simultaneously encourage consistency between predicted and ground truth labels and encode traditional label dependencies (instance similarity and class co-occurrence). Semantic hierarchy constraints are incorporated as hard linear constraints in the matrix optimization. The discrete problem is further relaxed to a continuous convex problem, which can be solved using ADMM [3]. We summarize our contributions next.

Contributions: **(1)** We address the MLML problem by using a mixed graph to encode a network of label dependencies: instance similarity, class co-occurrence, and semantic hierarchy. **(2)** Learning on this mixed graph is formulated as a linearly constrained convex matrix optimization problem that is amenable to efficient solvers. **(3)** Our extensive experiments on the tasks of image and video annotation show the superiority of our method in comparison to the state-of-the-art. **(4)** We augment labeling of several widely used datasets, including Corel5k [13], ESP Game [34], IAPRTC-12 [21] and MediaMill [30], with a fully labeled semantic hierarchy drawn from Wordnet [14]. This ground truth augmentation will be made publicly available to enable further research on the MLML problem in computer vision.

2. Related work

In the literature of multi-label learning, the previous works that are designed to handle the missing labels can be generally partitioned into four categories. First, the missing labels are directly treated as negative labels, including [8, 31, 5, 9, 1, 36, 37, 10]. Common to these methods is that the label bias is brought into the objective function. As a result, their performance is greatly affected when massive ground-truth positive labels are initialized as negative labels. Second, filling in missing labels is treated as a matrix completion (MC) problem, including [20, 6, 43]. A recent work called LEML [45] is proposed in the empirical risk minimization (ERM) framework. Both MC models and LEML are based on the low rank assumption. Although the low rank assumption is widely used, it may not hold in practical multi-label problem. Third, missing labels can be treated as latent variables in probabilistic models, including the model based on Bayesian network [24, 33] and conditional restricted Boltzmann machines (CRBM) model. Last, Wu et al. [39] define three label states, including positive labels $+1$, negative labels -1 and missing labels 0, to avoid the label bias. However, the two solutions proposed in [39] involves matrix inversion, which limits the scalability to handle larger datasets. Wu et al. [41] propose an inductive model based on the framework of regularized logistic regression. It also adopts three label states and a hinge loss function to avoid the label bias. However, the classifier parameters corresponding to each class have to be learned sequentially. Furthermore, the computational cost of this method increases significantly with the number of classes and becomes prohibitive for very large datasets.

Hierarchical multi-label learning (HML) [19] has been applied to problems where the label hierarchy exists, such as image annotation [32], text classification [28, 29] and protein function prediction [2, 44]. Except for a few cases, most existing HML methods only consider the learning problem of complete hierarchical labels. However, in real problems, the incomplete hierarchical labels commonly occur, such as in image annotation. Yu at al. [44] recently propose a method to handle the incomplete hierarchical labels. However, the semantic hierarchy and the multi-label learning are used seperately, such that the semantic hierarchy constraint can not be fully satisfied. Deng et al. [11] develops a CRF model for object classification. The semantic hierarchy constraint and missing labels are also incorporated in this model. However, a significant difference is that [11] focuses on a single object in each instance, while there are multiple object in each instance in our problem.

In the applications of image annotation and video annotation, both missing labels and semantic hierarchy have been explored in many previous works, such as [31, 5, 9, 42, 26, 16] (missing labels) and [32] (semantic hierarchy). However, to the best of our knowledge, no previous work in image and video annotation has extensively studied missing labels and semantic hierarchy simultaneously.

Note that the semantic hierarchy constraint used in our model is similar to the ranking constraint [15, 5] that is widely used in multi-label ranking models, but there are significant differences. Firstly, the ranking constraint used in these models means the predicted value of the provided positive label should be larger than the one of the provided negative label, while the semantic hierarchy constraint involves the ranking between the parent and the child class.

Besides, the ranking constraint is always incorporated as the loss function, while the semantic hierarchy constraint is formulated as the linear constraint in our model.

3. Problem and proposed model

3.1. Problem definition

Similar to traditional MLML problems, our method takes as input two matrices: a data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, which aggregates the d-dimensional feature vectors of all n (training and test) instances, and a label matrix $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \{0, \frac{1}{2}, +1\}^{m\times n}$, which aggregates the m-dimensional label vectors of the instances. Therefore, each instance $\mathbf{x}_i$ can take one or more labels from the m different classes $\{c_1, \ldots, c_m\}$. Its corresponding label vector $\mathbf{y}_i = \mathbf{Y}_{\cdot i}$ determines its membership to each of these classes. For example, if $\mathbf{Y}_{ji} = +1$, then $\mathbf{x}_i$ is a member of c_j and if $\mathbf{Y}_{ji} = 0$, then $\mathbf{x}_i$ is not a member of this class. However, if $\mathbf{Y}_{ji} = \frac{1}{2}$, then the membership of $\mathbf{x}_i$ to c_j is considered missing (i.e. it has a missing label). With this notation, all m labels of each testing instance $\mathbf{x}_k$ are missing, i.e. $\mathbf{y}_k = \frac{1}{2}\mathbf{1}$. The semantic hierarchy is encoded as another matrix: $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{n_e}] \in \mathbb{R}^{m\times n_e}$, with n_e being the number of directed edges. $\boldsymbol{\phi}_i = [0, \ldots, 1, \ldots, -1, \ldots, 0]^\top$ denotes the index vector of the ith directed edge (see Figure 1), with $\boldsymbol{\phi}_i(i_{parent}) = 1$ and $\boldsymbol{\phi}_i(i_{child}) = -1$, while all other entries are 0.

Our goal is to obtain a complete label matrix $\mathbf{Z} \in \{+1, 0\}^{m\times n}$ that satisfies the following three properties simultaneously. (i) $\mathbf{Z}$ is sufficiently consistent with the provided (not missing) labels in $\mathbf{Y}$, i.e. $\mathbf{Z}_{ij} = \mathbf{Y}_{ij}$ if $\mathbf{Y}_{ij} \neq \frac{1}{2}$. (ii) $\mathbf{Z}$ benefits from label similarity among similar instances and class co-occurrence, i.e. if $\mathbf{x}_i$ and $\mathbf{x}_j$ have similar features or the classes they belong to co-occur, then their corresponding predicted labels should be similar. (iii) $\mathbf{Z}$ is consistent with the semantic hierarchy $\mathbf{\Phi}$. To enforce this, we ensure that if c_a is the parent of c_b, a hard constraint is applied, which guarantees that the score (the presence probability) of c_a should not be smaller than the score of c_b. This constraint ensures that the final predicted labels are consistent with the semantic hierarchy. By incorporating all three criteria simultaneously, label information is propagated from provided labels to the missing labels.

3.2. Label consistency

Label consistency of $\mathbf{Z}$ with $\mathbf{Y}$ is formulated as

$$\ell(\mathbf{Y}, \mathbf{Z}) = \sum_{i,j}^{n,m} \overline{\mathbf{Y}}_{ij}(\mathbf{Y}_{ij} - \mathbf{Z}_{ij}) = \text{const} - \text{tr}(\overline{\mathbf{Y}}^\top \mathbf{Z}), \quad (1)$$

where const $= \text{tr}(\overline{\mathbf{Y}}^\top \mathbf{Y})$, and $\overline{\mathbf{Y}}$ is defined as $\overline{\mathbf{Y}}_{ij} = (2\mathbf{Y}_{ij} - 1) * \tau_{ij}$, where τ_{ij} is a penalty function mismatches between $\mathbf{Y}_{ij}$ and $\mathbf{Z}_{ij}$. We set τ_{ij} in the following manner. If $\mathbf{Y}_{ij} = 0$, then $\tau_{ij} = r_- > 0$, if $\mathbf{Y}_{ij} = +1$, then $\tau_{ij} = r_+ > r_-$, and if $\mathbf{Y}_{ij} = \frac{1}{2}$, then $\tau_{ij} = 0$. In doing so and unlike the work in [39, 41], a higher penalty is incurred if a ground truth label is $+1$ and predicted as 0, as compared to the reverse case. This idea embeds the observation that most entries of $\mathbf{Y}$ in many multi-label datasets (with a relatively large number of classes) are 0 and that $+1$ labels are rare. Of course, missing labels are not penalized. Eq (1) satisfies the label consistency. When $\mathbf{Y}_{ij} = +1$, $\mathbf{Z}_{ij}$ is encouraged to be $+1$; when $\mathbf{Y}_{ij} = 0$, $\mathbf{Z}_{ij}$ is encouraged to be 0; when $\mathbf{Y}_{ij} = \frac{1}{2}$, there is no constraint on $\mathbf{Z}_{ij}$.

3.3. Instance-level label dependency

Similar to [39, 41], we incorporate instance-level label similarity, i.e. property (ii), using the regularization term in Eq (2).

$$\text{tr}(\mathbf{Z}\mathbf{L}_\mathbf{X}\mathbf{Z}^\top) = \sum_{k,i,j}^{m,n,n} \frac{\mathbf{W}_\mathbf{X}(i,j)}{2}\left[\frac{\mathbf{Z}_{ki}}{\sqrt{\mathbf{d}_\mathbf{X}(i)}} - \frac{\mathbf{Z}_{kj}}{\sqrt{\mathbf{d}_\mathbf{X}(j)}}\right]^2, \quad (2)$$

where instance similarity matrix $\mathbf{W}_\mathbf{X}$ is: $\mathbf{W}_\mathbf{X}(i,j) = \exp -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\varepsilon_i \varepsilon_j}$, $\forall i \neq j$ and $\mathbf{W}_\mathbf{X}(i,i) = 0$. The kernel size $\varepsilon_i = \|\mathbf{x}_i - \mathbf{x}_h\|_2$ and $\mathbf{x}_h$ is the h-th nearest neighbor of $\mathbf{x}_i$. Similar to [39], we set $h = 7$. The normalization term $\mathbf{d}_\mathbf{X}(i) = \sum_j^n \mathbf{W}_\mathbf{X}(i,j)$ makes the regularization term invariant to the different scaling factors of the elements of $\mathbf{W}_\mathbf{X}$ [35]. The normalized Laplacian matrix is $\mathbf{L}_\mathbf{X} = \mathbf{I} - \mathbf{D}_\mathbf{X}^{-\frac{1}{2}}\mathbf{W}_\mathbf{X}\mathbf{D}_\mathbf{X}^{-\frac{1}{2}}$ with $\mathbf{D}_\mathbf{X} = \text{diag}(\mathbf{d}_\mathbf{X}(1), \cdots, \mathbf{d}_\mathbf{X}(n))$.

3.4. Class-level label dependency

Here, we consider two types of class-level label dependency, namely class co-occurrence and semantic hierarchy.

Class co-occurrence: This dependency is encoded using the regularization term in Eq (3).

$$\text{tr}(\mathbf{Z}^\top\mathbf{L}_\mathbf{C}\mathbf{Z}) = \sum_{k,i,j}^{n,m,m} \frac{\mathbf{W}_\mathbf{C}(i,j)}{2}\left[\frac{\mathbf{Z}_{ik}}{\sqrt{\mathbf{d}_\mathbf{C}(i)}} - \frac{\mathbf{Z}_{jk}}{\sqrt{\mathbf{d}_\mathbf{C}(j)}}\right]^2. \quad (3)$$

Here, we define the class similarity matrix $\mathbf{W}_\mathbf{C}$ as: $\mathbf{W}_\mathbf{C}(i,j) = \frac{<\mathbf{Y}_{i\cdot}, \mathbf{Y}_{j\cdot}>}{\|\mathbf{Y}_{i\cdot}\|\cdot\|\mathbf{Y}_{j\cdot}\|}$, $\forall i \neq j$ and $\mathbf{W}_\mathbf{C}(i,i) = 0$. The normalized Laplacian matrix is defined as $\mathbf{L}_\mathbf{C} = \mathbf{I} - \mathbf{D}_\mathbf{C}^{-\frac{1}{2}}\mathbf{W}_\mathbf{C}\mathbf{D}_\mathbf{C}^{-\frac{1}{2}}$ with $\mathbf{D}_\mathbf{C} = \text{diag}(\mathbf{d}_\mathbf{C}(1), \cdots, \mathbf{d}_\mathbf{C}(m))$.

Semantic hierarchy: To enforce the semantic hierarchy constraint, i.e. property (iii), we apply the following constraint: $\mathbf{Z}(i_{parent}, j) \geq \mathbf{Z}(i_{child}, j)$, $\forall i = 1, \ldots, n_e, \forall j = 1, \ldots, n$. The resulting constraints can be aggregated in matrix form, as

$$\mathbf{\Phi}^\top\mathbf{Z} \geq 0. \quad (4)$$

Algorithm 1 *ADMM for the proposed ML-MG problem*

Input: data matrix $\mathbf{X}$, initial label matrix $\mathbf{Y}$, Laplacian matrices $\mathbf{L_X}$ and $\mathbf{L_C}$, parameters β, γ and ρ

Output: predicted label matrix $\mathbf{Z}$

Initialize $\mathbf{Z}_0 = \mathbf{Y}$, $\mathbf{Z}_0$ as: if $\mathbf{Y}(i,j) \neq 0.5$, then set $\mathbf{Z}_0(i,j) = \mathbf{Y}(i,j)$, otherwise set $\mathbf{Z}_0(i,j) = 0$; $\mathbf{Q} = \mathbf{\Lambda} = \mathbf{0}$, and $t = 1$
while not converge **do**
  Update $\mathbf{Z}_{t+1}$ according to Eq (8);
  Update $\mathbf{Q}_{t+1}$ according to Eq (9);
  Update $\mathbf{\Lambda}_{t+1}$ according to Eq (10);
  Set $t = t + 1$; $\rho = 5\rho$
end while
return $\mathbf{Z}^* = \mathbf{Z}_{t+1}$.

3.5. MLML with mixed graph (ML-MG)

We formulate the MLML problem using the constructed mixed graph as a binary matrix optimization problem by linearly combining Eqs (1,2,3) to form the objective and Eq (4) to enforce the semantic hierarchy constraints.

$$\begin{aligned} \arg\min_{\mathbf{Z}} \quad & -\mathrm{tr}(\overline{\mathbf{Y}}^\top \mathbf{Z}) + \beta \mathrm{tr}(\mathbf{Z}\mathbf{L_X}\mathbf{Z}^\top) + \gamma \mathrm{tr}(\mathbf{Z}^\top \mathbf{L_C}\mathbf{Z}), \\ \text{s.t.} \quad & \mathbf{Z} \in \{0,1\}^{m\times n}, \quad \mathbf{\Phi}^\top \mathbf{Z} \geq 0. \end{aligned} \tag{5}$$

Due to the binary constraint on $\mathbf{Z}$, it is difficult to efficiently solve this discrete problem. Thus, we use a conventional *box* relaxation, which relaxes $\mathbf{Z}$ to take on values in $[0,1]^{m\times n}$. The relaxed ML-MG problem in Eq (5) is a convex quadratic problem (QP) with linear matrix constraints (refer to the **supplementary material** for more details).

$$\begin{aligned} \arg\min_{\mathbf{Z}} \quad & -\mathrm{tr}(\overline{\mathbf{Y}}^\top \mathbf{Z}) + \beta \mathrm{tr}(\mathbf{Z}\mathbf{L_X}\mathbf{Z}^\top) + \gamma \mathrm{tr}(\mathbf{Z}^\top \mathbf{L_C}\mathbf{Z}), \\ \text{s.t.} \quad & \mathbf{Z} \in [0,1]^{m\times n}, \quad \mathbf{\Phi}^\top \mathbf{Z} \geq 0. \end{aligned} \tag{6}$$

Due to its convexity and smoothness, the ML-MG problem can be efficiently solved by many solvers. In this work, we adopt the alternative direction of method of multipliers (ADMM) [3] for this purpose, since it is known to have attractive computational properties that have been recently exploited to tackle other popular problems in computer vision, including object tracking [46, 48], image classification [47], and image registration [18].

4. ADMM for ML-MG

After adding a non-negative slack variable $\mathbf{Q} \in \mathbb{R}^{n_e \times n}$, we can formulate the augmented Lagrange function of Problem (6) as in Eq (7), where $\mathbf{Z} \in [0,1]^{m\times n}$ and $\mathbf{Q} \geq 0$.

$$\begin{aligned} L_\rho(\mathbf{Z},\mathbf{Q},\mathbf{\Lambda}) &= -\mathrm{tr}(\overline{\mathbf{Y}}^\top \mathbf{Z}) + \beta \mathrm{tr}(\mathbf{Z}\mathbf{L_X}\mathbf{Z}^\top) \\ &+ \gamma \mathrm{tr}(\mathbf{Z}^\top \mathbf{L_C}\mathbf{Z}) + \mathrm{tr}[\mathbf{\Lambda}^\top(\mathbf{\Phi}^\top \mathbf{Z} - \mathbf{Q})] + \frac{\rho}{2}||(\mathbf{\Phi}^\top \mathbf{Z} - \mathbf{Q})||_F^2 \end{aligned} \tag{7}$$

Here, $\mathbf{\Lambda} \in \mathbb{R}^{n_e \times n}$ is the Lagrange multiplier (dual variable), $\rho > 0$ is a tradeoff parameter, and $||\cdot||_F$ denotes the Frobenius matrix norm. Following the conventional ADMM framework [3], we can minimize Problem (6) by alternating index. When updating $\mathbf{Z}_t$ in Eq (8), we set $\overline{\mathbf{A}}_t = -\overline{\mathbf{Y}} + \mathbf{\Phi}\mathbf{\Lambda}_t - \rho\mathbf{\Phi}\mathbf{Q}_t$, $\overline{\mathbf{B}}_t = \beta\mathbf{L_X}$ and $\overline{\mathbf{C}}_t = \gamma\mathbf{L_C} + \frac{\rho}{2}\mathbf{\Phi}\mathbf{\Phi}^\top$. The resulting problem is a convex QP with box constraints that can be efficiently solved using projected gradient descent (PGD) with exact line search [4]. Moreover, the updates for $\mathbf{Q}_t$ and $\mathbf{\Lambda}_t$ are closed form. More details of the optimization are provided in the **supplementary material**.

$$\begin{aligned} \mathbf{Z}_{t+1} &= \arg\min_{\mathbf{Z}\in[0,1]^{m\times n}} L_\rho(\mathbf{Z},\mathbf{Q}_t,\mathbf{\Lambda}_t) \\ &= \arg\min_{\mathbf{Z}\in[0,1]^{m\times n}} \mathrm{tr}[\overline{\mathbf{A}}_t^\top \mathbf{Z}] + \mathrm{tr}[\mathbf{Z}\overline{\mathbf{B}}_t\mathbf{Z}^\top] + \mathrm{tr}[\mathbf{Z}^\top\overline{\mathbf{C}}_t\mathbf{Z}] \end{aligned} \tag{8}$$

$$\begin{aligned} \mathbf{Q}_{t+1} &= \arg\min_{\mathbf{Q}\geq 0} \quad L_\rho(\mathbf{Z}_{t+1},\mathbf{Q},\mathbf{\Lambda}_t) \\ &= \max(0, \mathbf{\Phi}^\top\mathbf{Z}_{t+1} + \frac{1}{\rho}\mathbf{\Lambda}_t^\top) \end{aligned} \tag{9}$$

$$\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t + \rho[\mathbf{\Phi}^\top\mathbf{Z}_{t+1} - \mathbf{Q}_{t+1}]. \tag{10}$$

The ADMM solution to the ML-MG problem is summarized in Algorithm (1). Based on the proof in [17, 27], our ADMM algorithm is guaranteed to converge to the global minimum of Problem (6).

5. Experiments

In this section, we evaluate the proposed method and the state-of-the-art methods on four benchmark datasets in image annotation and video annotation.

5.1. Experimental setup

Datasets. Four benchmark multi-label datasets are used in our experiments, including Corel5k [13], ESP Game [34], IAPRTC-12 [21], and MediaMill [30]. These datasets are chosen because they are representative and popular benchmarks for comparative analysis among MLML methods. Since the scope of this paper is not feature design, we obtain the data and label matrices ($\mathbf{X}$ and $\mathbf{Y}$) of the first three image datasets from the seminal work [22] [1]. Each image in these datasets is described by the dense SIFT features and is represented by a 1000-dimensional vector. The features and labels of the video dataset MediaMill are downloaded from the 'Mulan' website [2].

Semantic hierarchy. We build a semantic hierarchy for each dataset based on Wordnet [14]. Specifically, for each dataset, we search for each class in Wordnet and extract one or more directed paths (i.e., a long sequence of directed edges from parent class to child class). In each path, we identify the nearest class that is also in the label vocabulary

[1] http://lear.inrialpes.fr/people/guillaumin/data.php
[2] http://mulan.sourceforge.net/datasets-mlc.html

(i.e. the set $\{c_1, \cdots, c_m\}$ of all classes of this dataset) as the parent class. This procedure is repeated for all m classes in the dataset to form the semantic hierarchy matrix $\mathbf{\Phi}$. In the same manner, we build the hierarchy for each of the four datasets. Similar to [32], we also consider two types of semantic dependency: "is a" and "is a part of". For example, a part of the semantic hierarchy of Corel 5k is shown in Figure 2. Due to the space limit, we provide the complete semantic hierarchies and the complete label matrices for all four datasets in the **supplementary material**. A summary of these hierarchies are given in Table 1.

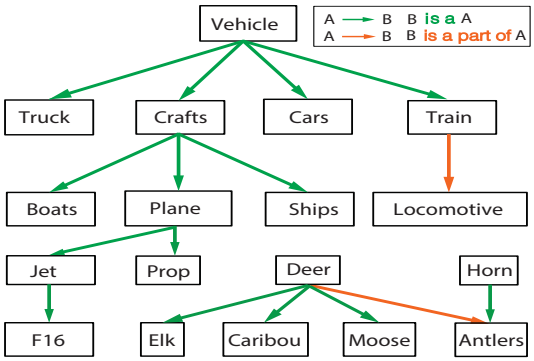


Figure 2: A part of the semantic hierarchy of Corel 5k

dataset	# nodes	# edges	# root	# leaf	# singleton	depth
Corel 5k [13]	260	138	37	98	99	5
ESP Game [34]	268	129	41	92	120	4
IAPRTC-12 [21]	291	179	36	132	98	4
MediaMill [30]	101	63	14	52	30	3

Table 1: *Details of the semantic hierarchies for the different datasets that we augmented.*

Note that in all the datasets, the provided ground-truth label matrices do not fully satisfy the semantic hierarchy constraints. In other words, some images or videos are labeled with a child class but not with the corresponding parent class. Therefore, we augment the label matrix according to the semantic hierarchy for each dataset. The semantically enhanced comprehensive ground-truth label matrix is referred to as "complete", while the original label matrix as "original". The basic statistics of both the complete and original label matrix are summarized in Table 2.

Methods for comparison. Several state-of-the-art and recent multi-label methods that can handle missing labels are used for comparison, including MLR-GL [5], MC-Pos [6], FastTag [9], MLML-exact and MLML-appro [39] and LEML [45]. MLR-GL and FastTag are specially developed for image annotation, while other methods are general machine learning methods. Also, a state-of-the-art method in hierarchical multi-label learning, CSSAG [2], is also evaluated. CSSAG is a decoding method based on the predicted label matrix of one another algorithm, i.e., the kernel dependency estimation (KDE) algorithm [38]. However, the KDE algorithm doesn't work in the case of missing labels. To make a fair comparison between CSSAG and ML-MG, the results of ML-PGD are used as the input of CSSAG. The results are obtained with publicly available MATLAB source code of these methods provided by the authors. As a baseline, we also compare with a binary SVM classifier[3], which is trained on only labeled instances of each class.

Evaluation metrics. Average precision (AP) [49] is adopted to measure the ranking performance of the continuous label matrix $\mathbf{Z}$ generated by each of the 8 methods. Top-5 accuracy is also used to measure the predicted discrete labels (the top 5 ranked labels of each instance are set as positive, while others are negative labels). To quantify the degree to which the semantic hierarchy constraints are violated, we adopt a simplified hierarchical Hamming loss, similar to [28],

$$\ell_H^k(\hat{\mathbf{Z}}_k, \mathbf{Y}_C) = \frac{1}{nm} \sum_{i,j}^{n,m} \mathbb{I}\big[(\hat{\mathbf{Z}}_k(i,j) = 1) \wedge (\hat{\mathbf{Z}}_k(pa(i),j) = 0) \wedge (\mathbf{Y}_C(pa(i),j) = 0)\big], \tag{11}$$

where $\hat{\mathbf{Z}}_k$ denotes the discrete label matrix generated by setting the top-k labels in the continuous label vector of each instance as $+1$, while all others as 0. $\mathbf{Y}_C$ denotes the complete ground-truth label matrix. Then we define an *average hierarchical loss* (AHL) as $\overline{\ell}_H = \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \ell_H^k$. We set $\mathcal{S} = \{5, 10, 20, 50, 100, 150\}$ in our experiments.

Other settings. To simulate different scenarios with missing labels, we create training datasets with varying portions of provided labels, ranging from 5% (i.e., 95% of the whole training label matrix is missing) to 100% (i.e., no missing labels). In each case, the missing labels are randomly chosen among leaf and singleton classes[4] in the semantic hierarchy, and are set to $\frac{1}{2}$ in the original label matrix of the training data. We repeat this process 5 times to obtain different data splits. In all cases, the experimental results of test data are computed based on the complete label matrix. The reported results are summarized as the mean and standard deviation over all the runs. The average runtime of each method on the different datasets is also reported. Because the authors of [45] only provide the MEX file for LEML under Ubuntu system, we run it on a workstation with Ubuntu 12.04.2 and Intel Xeon X5650 2.67 GHz CPU. All other methods are run on the same machine with Windows 7. The trade-off parameters β and γ are tuned by cross-validation. The tuning ranges are set as $\beta \in \{0.1, 1, 5, 10, 50\}$ and $\gamma \in \{0, 0.01, 0.1, 1, 10\}$. Both $\mathbf{W_X}$ and $\mathbf{W_C}$ are defined as sparse matrices. The numbers of neighbors of each in-

[3] Trained with the LIBSVM package [7].

[4] If missing labels are generated on root and intermediate classes, many of them can be directly inferred as positive labels if their children classes are positive. Consequently the true missing label proportion may be inconsistent with the original designed proportion. To avoid this variation, we only generate missing labels on leaf and singleton classes.

dataset	# instances (training, test)	# class	# feature	$k_X, k_C, \frac{r_+}{r_-}$	label matrix	avg pos-class/inst	avg inst/pos-class	pos-class rate
Corel 5k [13]	4999 = 4500 + 499	260	1000	20, 10, 100	original	3.40	65.30	1.31%
					complete	4.84	93.06	1.86%
ESP Game [34]	20770 = 18689 + 2081	268	1000	20, 10, 100	original	4.69	363.2	1.75%
					complete	7.27	563.6	2.71%
IAPRTC-12 [21]	19627 = 17665 + 1962	291	1000	20, 10, 100	original	5.72	385.71	1.97%
					complete	9.88	666.3	3.39%
MediaMill [30]	43907 = 30993 + 12914	101	120	20, 10, 100	original	4.38	1902	4.33%
					complete	6.17	2680	6.10%

Table 2: *Data statistics of features and label matrices of four benchmark datasets.*

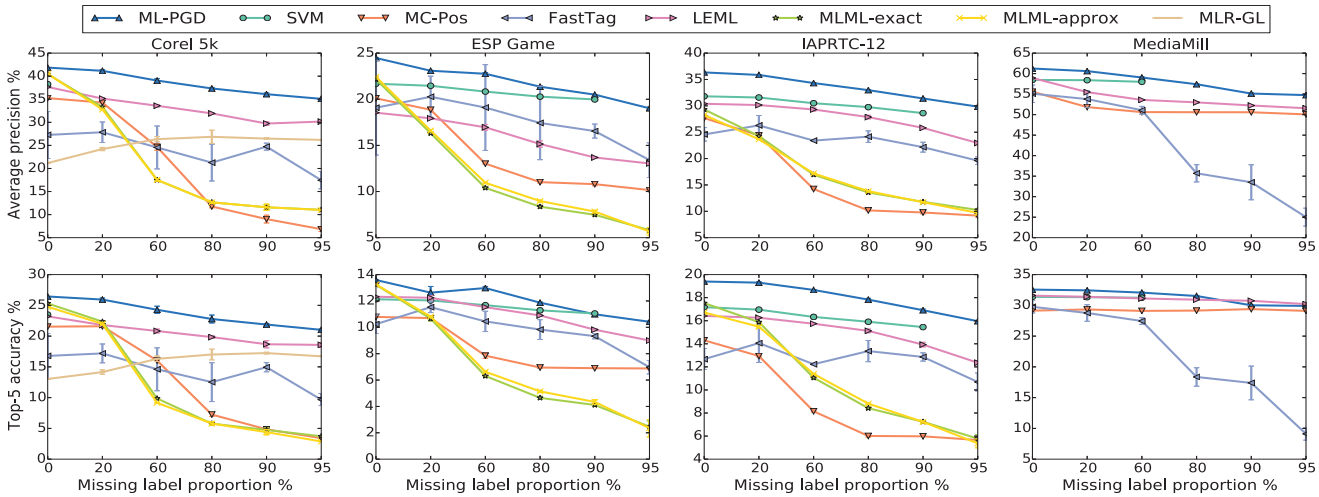


Figure 3: Average precision (**top**) and top-5 accuracy (**bottom**) results of four benchmark datasets for methods without semantic hierarchy. The bar on each point indicates the corresponding standard deviation. Figure better viewed on screen.

stance/class k_X and k_C are set as 20 and 10 respectively.

5.2. Results without semantic hierarchy

Figure 3 shows AP and top-5 accuracy results of the eight methods when the semantic hierarchy is not used, i.e., when $\mathbf{\Phi} = \mathbf{0}$. In this case, the inequality constraints in ML-MG are degenerate. Then the proposed model is a convex QP with box constraints that is solvable using the PGD method. We denote the hierarchy-free version of ML-MG as ML-PGD. ML-PGD consistently outperforms the other methods, thus, showing its superiority over other MLML methods even without hierarchy information. The improvement over the most competitive method on the four datasets is usually 5% (AP) or 3% (accuracy). Compared with MLML-exact and MLML-approx, ML-PGD shows significant improvement, especially when large proportions of the labels are missing. This is due to two main reasons. Firstly, there are many noisy negative labels in the original training label matrix, i.e., some positive labels 1 are set to 0. Since a larger penalty is incurred when misclassifying a positive label in ML-PGD, the influence of noisy negative labels can be alleviated. However, this is not the case for both MLML-exact and MLML-approx. Secondly, ML-PGD does not give any bias to missing labels. In contrast, missing labels are encouraged to be intermediate values between negative and positive labels in MLML-exact and MLML-approx, which brings in label bias. This is why their performance decreases significantly as the missing proportion increases. Interestingly, most of the multi-label methods are outperformed by the binary SVM in most cases. This suggests that they are sensitive to noisy and missing labels. Also, we note that some of the baseline methods fail to produce results in some test cases, especially when the missing label proportion is high. For example, SVM fails when there are no positive instances for some classes. Because of its extremely slow runtime ($> 10^4$ seconds per iteration), it was infeasible to run MLR-GL on the last three datasets. Similarly, the high memory requirements of MLML-exact and MLML-approx preclude running them on MediaMill data.

5.3. Results with semantic hierarchy

The results of utilizing the semantic hierarchy are shown in Figure 4, where the AP results of ML-PGD are repeated to facilitate the comparison.

ML-PGD vs. **ML-MG**. In the case of 0% missing labels, the improvement of ML-MG over ML-PGD ranges from $10\% - 19\%$ across the four datasets. This suggests that the semantic hierarchy constraints provide very useful informa-

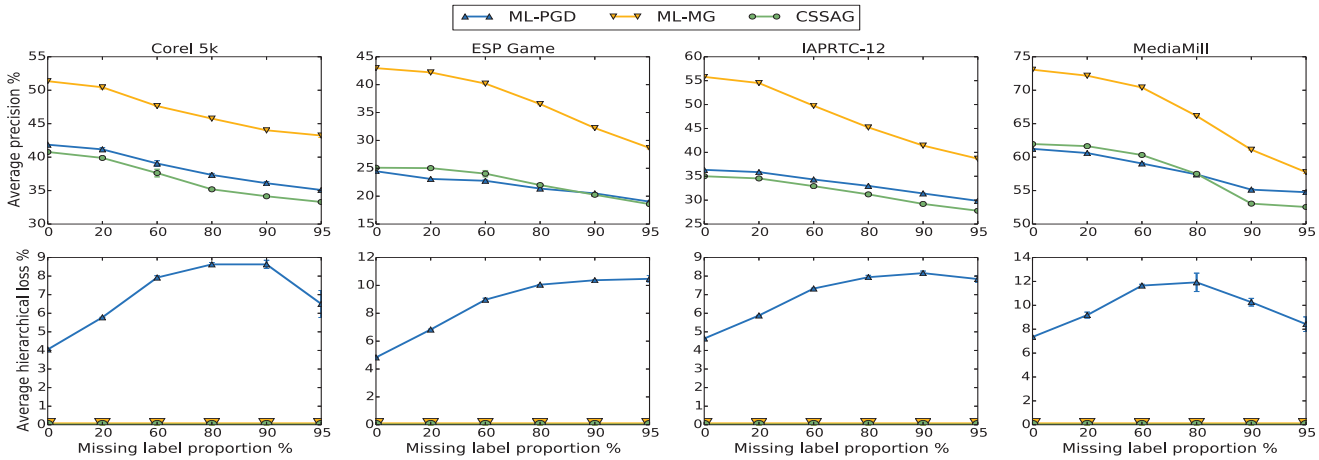


Figure 4: Average precision (AP) and average hierarchical loss (AHL) results with semantic hierarchy.

tion to correct the noisy negative labels. Also, ML-MG significantly outperforms ML-PGD at high missing label proportions. This lends evidence to the fact that instance-level similarity and class co-occurrence are not enough to regularize the MLML problem and that semantic hierarchy provides essential information to fill in missing labels.

CSSAG vs. **ML-MG**. We take the continuous labels generated by ML-PGD as the input to CSSAG, which will one of two processing choices: one is to change the input continuous label score to 0 according to the semantic hierarchy and the predefined number of positive labels, while the other is to keep it unchanged. Thus, the AP results of its continuous outputs are similar to those of its input label matrix, i.e., ML-PGD. Although CSSAG can ensure that there are no inconsistent labels in its discrete label matrix (setting all positive continuous labels as discrete positive labels), it cannot provide a consistent continuous label ranking. In contrast, ML-MG can satisfy these two conditions simultaneously.

Qualitative Results. Figure 5 shows labels predicted by our proposed ML-MG method when it is applied to the task of image annotation. It is worthwhile to note that parent classes are always ranked ahead of their children classes, thus, visualizing the direct effect of semantic hierarchy. Due to the space limit, more qualitative examples are provided in the **supplementary material**.

5.4. Convergence and sensitivity to initialization

Here, we evaluate the convergence of ML-MG using different label initializations. We only present the curve in the case of 0% missing labels, as shown in Figure 6. In the top row, we initialize the label matrix $\mathbf{Z}$ by setting all missing labels as 0. In this case, ML-MG converges to its best AP in less than 30 iterations on Corel 5k and in less than 10 on the other datasets. In the bottom row, missing labels are initialized as random values in $[0, 1]$. We repeat the random initializations 10 times and report mean and std values. The extremely small std values of both objective function and AP values suggest ML-MG is insensitive to initialization.



Figure 5: Some prediction results of our ML-MG method. The images in the top row are extracted from ESP Game [34], while the other two rows from IAPRTC-12 [21]. The predicted labels are ranked in descending order according to their label scores.

5.5. Computational analysis

The computational complexity of ML-PGD is $O_{PGD} = O(T_1((2k_X + k_C)mn + 3k_Cm^2 + 8m^2n))$, while that of ML-MG is $O_{MG} = O(T_2(O_{PGD} + 2n_emn))$. T_1 and T_2 denote the number of PGD and ADMM iterations respectively. As shown in Figure 6, T_2 is always very small. The detailed derivation of the complexity is provided in the **supplementary material**. To emphasize the computational efficiency of our method, we report the average runtime of all methods on the four datasets in Table 3. In the case of 0% missing labels, each method is run 10 times and the average

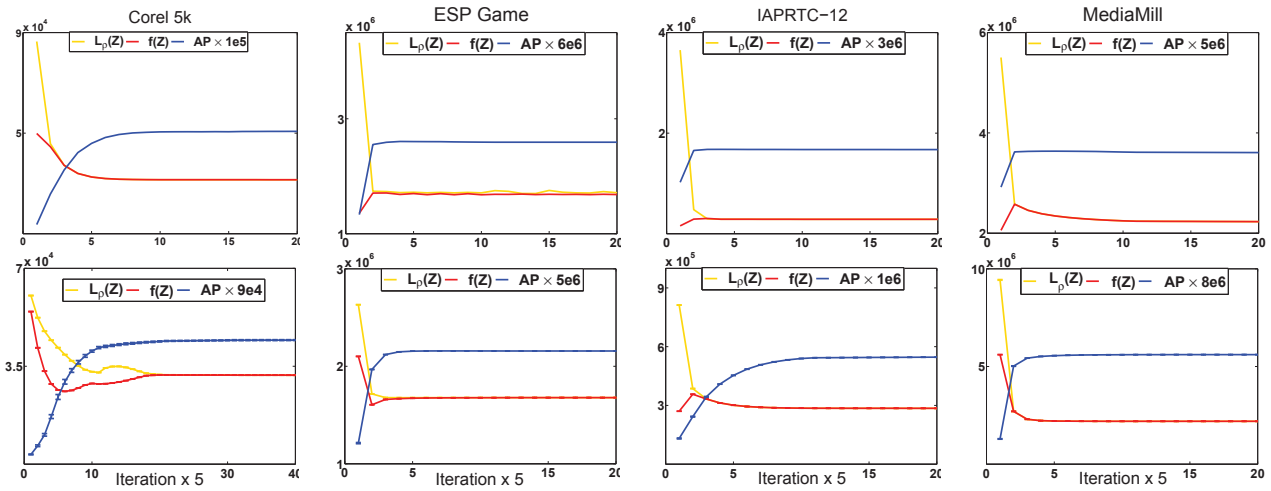


Figure 6: Convergence curve of ML-MG with (**Top**): initializing missing labels as 0; (**Bottom**): initializing missing labels as random (10 times) values in $[0, 1]$. $L_\rho(\mathbf{Z})$ and $f(\mathbf{Z})$ denote the objective functions of Eq (7) and (6) respectively. AP indicates the evaluation value of average precision. In the bottom row, as the std values are very small compared to the mean values, it is better to enlarge the figure to check them.

runtime is recorded. For MLR-GL, LEML and ML-MG, the number of maximum iterations is set as 20, while 50 for ML-PGD. The ranks of the mapping matrix in LEML are set as 50, 50, 50, and 20 for the four datasets, respectively. Clearly, our proposed ML-MG method (and its hierarchy-free variant ML-PGD) are significantly more computationally attractive than the other methods.

Datasets	SVM [7]	MLR-GL [5]	MC-Pos [6]	FastTag [9]	LEML [45]	MLML -exact [39]	MLML -appro [39]	ML -PGD	ML -MG
Corel5k	8826	1820	183.17	70.40	180.4	54.30	2.72	**0.83**	10.90
ESP Game	41817	–	662.8	201.2	595.6	2652.6	120	**4.13**	50.40
IAPRTC-12	35373	–	465	213	553.6	2263	98.30	**5.49**	52.32
MediaMill	6361	–	437.6	87.10	253.3	–	–	**7.25**	74.4

Table 3: Runtime in seconds of all 9 methods

6. Conclusions and discussions

This work proposes a novel model to handle the problem of multi-label learning with missing labels. A unified network of label dependency is built based on a mixed graph, which jointly incorporates instance-level label similarity and class co-occurrence as undirected edges, as well as, semantic hierarchy as directed edges. A convex problem is formulated by encoding the undirected edges as regularization terms, while embedding the directed edges as linear constraints. Thus multi-label learning and enforcing semantic hierarchy constraints can be performed simultaneously. A computationally attractive algorithm based on ADMM is used to exactly optimize this problem. Applying our method on image and video annotation tasks have demonstrated its superior performance against state-of-the-art methods. Moreover, we contribute manually generated semantic hierarchies for four popular benchmark datasets, which will be beneficial to the research community at large.

We realize that semantic hierarchy and missing labels exist in many other real-world problems, including text classification [28, 29], tracking [40, 48], action recognition [50, 51] and activity recognition [23]. In the future, we aim to apply ML-MG to these applications.

Acknowledgments: This work is supported supported by competitive research funding from King Abdullah University of Science and Technology (KAUST). The participation of Siwei Lyu in this work is partly supported by US National Science Foundation Research Grant (CCF-1319800) and National Science Foundation Early Faculty Career Development (CAREER) Award (IIS-0953373). We thank Fabian Caba Heilbron for his help on figure plotting, Rafal Protasiuk for his help on data collection, and Ganzhao Yuan for the discussion. We thank the reviewers for their constructive comments.

References

[1] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, pages 13–24, 2013. 2

[2] W. Bi and J. T. Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *ICML*, pages 17–24, 2011. 2, 5

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 2, 4

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 4

[5] S. S. Bucak, R. Jin, and A. K. Jain. Multi-label learning with incomplete class assignments. In *CVPR*, pages 2801–2808. IEEE, 2011. 2, 5, 8

[6] R. S. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Matrix completion for multi-label image classification. In *NIPS*, pages 190–198, 2011. 2, 5, 8

[7] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. 5, 8

[8] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SIAM international conference on data mining*, pages 410–419, 2008. 2

[9] M. Chen, A. Zheng, and K. Weinberger. Fast image tagging. In *ICML*, pages 1274–1282, 2013. 2, 5, 8

[10] Z. Chen, M. Chen, K. Q. Weinberger, and W. Zhang. Marginalized denoising for link prediction and multi-label learning. In *AAAI*, 2015. 2

[11] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, pages 48–64. Springer, 2014. 2

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009. 1

[13] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, pages 97–112. Springer, 2002. 2, 4, 5, 6

[14] C. Fellbaum. *WordNet*. Wiley Online Library, 1998. 2, 4

[15] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker. Multi-label classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008. 2

[16] B. Geng, L. Yang, C. Xu, and X.-S. Hua. Collaborative learning for image and video annotation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 443–450. ACM, 2008. 2

[17] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. 2013. 4

[18] B. Ghanem, T. Zhang, and N. Ahuja. Robust video registration applied to field-sports video analysis. In *International Conference on Acoustics, Speech and Signal Processing*, 2012. 4

[19] E. Gibaja and S. Ventura. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(6):411–444, 2014. 2

[20] A. B. Goldberg, X. Zhu, B. Recht, J.-M. Xu, and R. D. Nowak. Transduction with matrix completion: Three birds with one stone. In *NIPS*, pages 757–765, 2010. 2

[21] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The iapr tc-12 benchmark: A new evaluation resource for visual information systems. In *International Workshop OntoImage*, pages 13–23, 2006. 2, 4, 5, 6, 7

[22] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, pages 309–316, 2009. 4

[23] F. C. Heilbron, V. Castillo, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 8

[24] A. Kapoor, R. Viswanathan, and P. Jain. Multilabel classification using bayesian compressed sensing. In *NIPS*, pages 2654–2662, 2012. 2

[25] X. Li, F. Zhao, and Y. Guo. Conditional restricted boltzmann machines for multi-label learning with incomplete labels. In *AISTATS*, pages 635–643, 2015. 1

[26] Z. Lin, G. Ding, M. Hu, J. Wang, and X. Ye. Image tag completion via image-specific and tag-specific linear sparse reconstructions. In *CVPR*, pages 1618–1625, 2013. 2

[27] A. U. Raghunathan and S. Di Cairano. Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control,. In *Proceedings of Symposium on Mathematical Theory of Networks and Systems*, pages 807–814, 2014. 4

[28] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classification models. In *ICML*, pages 744–751. ACM, 2005. 2, 5, 8

[29] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *The Journal of Machine Learning Research*, 7:1601–1626, 2006. 2, 8

[30] C. G. Snoek, M. Worring, J. C. Van Gemert, J.-M. Geusebroek, and A. W. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430. ACM, 2006. 2, 4, 5, 6

[31] Y. Sun, Y. Zhang, and Z.-H. Zhou. Multi-label learning with weak label. In *AAAI*, pages 593–598, 2010. 2

[32] A.-M. Tousch, S. Herbin, and J.-Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012. 2, 5

[33] D. Vasisht, A. Damianou, M. Varma, and A. Kapoor. Active learning for sparse bayesian multilabel classification. In *SIGKDD*, pages 472–481. ACM, 2014. 2

[34] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM, 2004. 1, 2, 4, 5, 6, 7

[35] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 3

[36] Q. Wang, B. Shen, S. Wang, L. Li, and L. Si. Binary codes embedding for fast image tagging with incomplete labels. In *ECCV*, pages 425–439. Springer, 2014. 2

[37] Q. Wang, L. Si, and D. Zhang. Learning to hash with partial tags: Exploring correlation between tags and hashing bits for large scale image retrieval. In *ECCV*, pages 378–392. Springer, 2014. 2

[38] J. Weston, O. Chapelle, V. Vapnik, A. Elisseeff, and B. Schölkopf. Kernel dependency estimation. In *NIPS*, pages 873–880, 2002. 5

[39] B. Wu, Z. Liu, S. Wang, B.-G. Hu, and Q. Ji. Multi-label learning with missing labels. In *ICPR*, 2014. 1, 2, 3, 5, 8

[40] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Simultaneous clustering and tracklet linking for multi-face tracking in videos. In *ICCV*. IEEE, 2013. 8

[41] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji. Multi-label learning with missing labels for image annotation and facial action unit recognition. *Pattern Recognition*, 48(7):2279–2289, 2015. 1, 2, 3

[42] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *TPAMI*, 35(3):716–727, 2013. 2

[43] M. Xu, R. Jin, and Z.-H. Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, pages 2301–2309, 2013. 2

[44] G. Yu, H. Zhu, and C. Domeniconi. Predicting protein functions using incomplete hierarchical labels. *BMC bioinformatics*, 16(1):1, 2015. 2

[45] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, pages 593–601, 2014. 1, 2, 5, 8

[46] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In *ECCV*, pages 470–484. 2012. 4

[47] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja. Low-rank sparse coding for image classification. In *ICCV*, pages 281–288, 2013. 4

[48] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *IJCV*, 111(2):171–190, 2014. 4, 8

[49] Y. Zhang and Z.-H. Zhou. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data*, 4(3):14, 2010. 5

[50] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu. Action recognition using context-constrained linear coding. *Signal Processing Letters*, 19(7):439–442, 2012. 8

[51] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu. Cross-view action recognition using contextual maximum margin clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1663–1668, 2014. 8