Tag Taxonomy Aware Dictionary Learning for Region Tagging

Jingjing Zheng, Zhuolin Jiang
Center for Automatic Research
University of Maryland, College Park, MD, USA
{zjngjng, zhuolin}@umiacs.umd.edu

Abstract

Tags of image regions are often arranged in a hierarchical taxonomy based on their semantic meanings. In this paper, using the given tag taxonomy, we propose to jointly learn multi-layer hierarchical dictionaries and corresponding linear classifiers for region tagging. Specifically, we generate a node-specific dictionary for each tag node in the taxonomy, and then concatenate the node-specific dictionaries from each level to construct a level-specific dictionary. The hierarchical semantic structure among tags is preserved in the relationship among node-dictionaries. Simultaneously, the sparse codes obtained using the level-specific dictionaries are summed up as the final feature representation to design a linear classifier. Our approach not only makes use of sparse codes obtained from higher levels to help learn the classifiers for lower levels, but also encourages the tag nodes from lower levels that have the same parent tag node to implicitly share sparse codes obtained from higher levels. Experimental results using three benchmark datasets show that the proposed approach yields the best performance over recently proposed methods.

1. Introduction

Region tagging, whose goal is to assign image regions with labeled tags, has attracted significant attention in computer vision and multimedia [11, 25, 7, 26, 21, 22]. Region tagging at a more fine-grained region-level has two benefits. First, it establishes the correspondences between image regions and semantic labels and thus can handle the diversity and arbitrariness of Web image content well. Second, experiments in [3, 21] reveal that accurate region-level annotations can effectively boost the performance of image-level annotations. In order to achieve robust content-based image retrieval, we focus on improving the accuracy of region tagging.

Recently several proposed region tagging approaches attempt to explore the contextual constraints among image regions using sparse coding techniques [11, 25, 7]. However, these approaches that simply used all training regions as the dictionary for spare coding have three main disadvantages. First, redundancy in training regions can increase the reconstruction error, which may degrade the effectiveness of region tagging. Second, the computational complexity of sparse coding increases with the size of dictionary and it is impossible to use all the training regions as the dictionary for large-scale datasets. Thus learning a compact and discriminative dictionary for region tagging is desirable. Third, for datasets with unbalanced tag classes, the performance of these approaches may decrease drastically. This is because unbalanced tag classes result in an unbalanced group structure in the dictionary such that the computed sparse codes become less discriminative for classification task. In addition, tags are often arranged into a hierarchical taxonomy based on their semantic meanings, such as the tag taxonomy shown in Figure 1. However, the tag taxonomy has not been exploited to improve the accuracy of region tagging, even though the similar category taxonomy has been shown to benefit the accuracy as well as the scalability of learning algorithms [15, 16, 6] for object recognition.

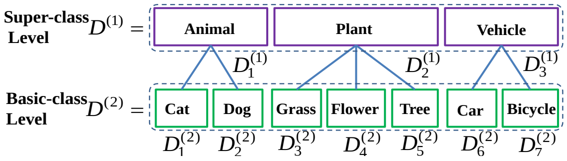


Figure 1. **A two-layer tag taxonomy and the corresponding dictionary framework.** This tag taxonomy has two levels: super-class level and basic-class level. At the super-class level, training samples are divided into three super-classes *Animal*, *Plant* and *Vehicle*, whereas training samples within each super-class are further divided into a few basic classes. We associate each tag node with a node-specific dictionary and concatenate the node-specific dictionaries from each level to create a level-specific dictionary. The level-specific dictionaries for this taxonomy are $D^{(1)}$ and $D^{(2)}$ while the node-specific dictionaries are $\{D_s^{(1)}\}_{s=1...3}$ and $\{D_k^{(2)}\}_{k=1...7}$. We reconstruct each image region using different level-specific dictionaries and sum up the sparse codes obtained from different levels as the final feature representation to learn a linear classifier for region tagging.

To overcome the above drawbacks, we present a novel multi-layer hierarchical dictionary learning framework for region tagging when the tag taxonomy is known. For illustration, a two-layer tag taxonomy and the corresponding dictionary learning framework is depicted in Figure 1. To our best knowledge, we are the first to use the supervised dictionary learning to explore the semantic relationship among tags. Specifically, we generate a node-specific dictionary for each tag node in the taxonomy and concatenate the node-specific dictionaries in each level to construct a level-specific dictionary. Thus the hierarchical semantic relationship among tags is preserved in the relationship among node-specific dictionaries, which enables us to exploit the discriminative information among regions in a hierarchial way. Moreover, dictionary items from the same node-specific dictionary are considered as a group so it introduces a group structure for each level-specific dictionary. Based on each level-specific dictionary and corresponding group structure, we reconstruct each image region using the group sparse coding algorithm [27] to obtain level-specific sparse codes. Compared with single-level sparse codes in existing sparse coding-based region tagging approaches [11, 25, 7], our multi-layer sparse codes not only encodes the contextual constraints among regions, but also encodes the relationship among tags. Finally, we sum up the sparse codes obtained from different levels as the final feature representation to learn a linear class classifier. For datasets with unbalanced tag classes, we can create balanced group structure for higher levels and make use of sparse codes obtained from higher levels to help design the classifiers for lower levels. Therefore, our approach is robust to datasets with unbalanced tag classes in contrast to existing sparse coding-based region tagging approaches that tend to perform poorly on datasets with unbalanced tag classes.

1.1. Our Contribution

The main contributions in our paper are four-fold:

- We present a multi-layer supervised dictionary learning framework that simultaneously learns multi-layer dictionaries and classifiers.
- We are the first to use the supervised dictionary learning to explore the semantic structure among tags, which not only takes advantages of the compactness and efficiency of dictionary learning, but also explores different group structures among image regions.
- Our approach proposes to sum up sparse codes from different levels as the feature representation to learn a linear classifier, which enables us to make use of discriminative information encoded in sparse codes from different levels.
- Our approach is robust to datasets with unbalanced tag classes.

2. Related Work

Recently, several region tagging approaches have used sparse coding techniques to encode contextual constraints among image regions for region tagging [11, 25, 7]. [11] proposed a bi-layer sparse coding framework to reconstruct image regions from over-segmented image patches that belong to a few images, and then propagate image labels of selected patches to the entire label to obtain region assignment. However, this method ignores the contextual correlations among regions, *e.g.*, co-occurrence and spatial correlations. [25] considered regions within the same image as a group, and used the group sparse coding with spatial kernels to jointly reconstruct image regions in the same image from other training regions. However, the contextual correlations of training regions across images are ignored due to the group structure of regions-in-image relationship. [7] extended group sparse coding with graph-guided fusion penalty to encourage highly correlated regions to be jointly selected for the reconstruction. However, the performance of the group sparse coding depends on a balanced group structure which has the similar number of training regions in each group so it might not be robust to datasets that have very unbalanced training regions.

Other techniques have also been proposed to boost the performance for region tagging or region-based image annotation. [21, 22] used multiple-instance learning techniques to learn the correspondence between image regions and keywords. The idea is that each image is annotated by the tag that has at least one sample region (seen as '*instance*') within this image (seen as '*bag*'). [26] regularized segmented image regions into $2D$ lattice layout, and employed a simple grid-structure graphical model to characterize the spatial context constraints. [3] used both the dominant image region and the relevant tags to annotate the semantics of natural scenes. [9] proposed a unified solution to tag refinement and tag-to-region assignment by using a multi-edge graph, where each vertex of the graph is a unique image encoded by a region bag with multiple image segmentations. [5] proposed a multi-layer group sparse coding framework to encode the mutual dependence between the class labels as well as the tag distribution information.

Supervised dictionary learning which combines dictionary learning with classifier training into a unified learning framework has been extensively studied [24, 17, 14, 28]. [24] performed supervised dictionary learning by minimizing the training error of classifying the image-level features, which are extracted by max pooling over the sparse codes within a spatial pyramid. [14] proposed a novel sparse representation of signals belonging to different classes in terms of a shared dictionary and discriminative models. This approach alternates between the step of sparse coding and the step of dictionary update and discriminative model learning. [28] extended the K-SVD algorithm by incorporating the classification error into an objective function that allows the simultaneous optimization of the dictionary and classifiers. In addition, [8, 1] proposed to use proximal methods for structured sparse learning where dictionary items are embedded in different structures.

3. Tag Taxonomy Aware Dictionary Learning

In this section, we first introduce the group sparse coding algorithm and then describe the formulation of our multi-layer supervised dictionary learning, its optimization and how to tag image regions using sparse codes.

3.1. Group Sparse Coding

Given a dictionary $D = [D_1, D_2, ..., D_G] \in \mathbb{R}^{d\times J}$ where $D_g \in \mathbb{R}^{d\times J_g}$ consists of a group of J_g visually correlated dictionary items, an image region $x \in \mathbb{R}^d$ can be reconstructed from the dictionary with the group LASSO penalty [27] as follows:

$$\begin{aligned}\mathbf{z} &= arg\min_{\mathbf{z}} \frac{1}{2}||x - \sum_{g=1}^{G} D_g z_g||_2^2 + \lambda \sum_{g=1}^{G} \beta_g ||z_g||_2 \\ &= arg\min_{\mathbf{z}} \frac{1}{2}||x - D\mathbf{z}||_2^2 + \lambda \sum_{g=1}^{G} \beta_g ||z_g||_2\end{aligned} \tag{1}$$

where $\mathbf{z} = [z_1^T, z_2^T, ..., z_G^T]^T \in \mathbb{R}^{J\times 1}$ is the reconstruction coefficients where z_g is the encoding coefficient corresponding to the g^{th} group. And $\lambda \geq 0$ is a trade-off parameter and $\beta_g = \sqrt{J_g}$ weights the penalty from the g-th group. Since the group LASSO uses a group-sparsity-inducing regularization instead of the l_1 norm as in LASSO [20], we can treat multiple visually similar dictionary items within the same group as a whole and exploit implicit relations among these dictionary items to some extent.

3.2. Multi-layer Supervised Dictionary Learning

We consider an image dataset $\mathcal{D}$ with a two-layer tag taxonomy whose levels from the top to the bottom are called: super-class level and basic-class level as shown in Figure 1. Note that extensions to learning multiple level-specific dictionaries for a multi-layer tag taxonomy can be accomplished in a similar way. Suppose that each image has been segmented into regions and a d-dimensional feature vector has been extracted for each region. Let $X \in \mathbb{R}^{d\times N}$ denote N training image regions from K tag classes. According to the tag taxonomy, image regions from these K classes in the basic-class level can be merged into S super-classes in the super-class level, *e.g.*, *cat* and *dog* belong to the super-class *animal* , whereas *grass* and *tree* belong to the super-class *plant* (See Figure 1). Thus each image region has one class label from the basic-class level and one super-class label from the super-class level. Let $H^{(2)} \in \{0, 1\}^{K\times N}$ denote the class label indicator matrix for all the regions, where $H^{(2)}_{(i,j)} = 1$ if the jth image region belongs to the ith tag and $H^{(2)}_{(i,j)} = 0$ otherwise. Similarly, we use $H^{(1)} \in \{0, 1\}^{S\times N}$ to denote the super-class label indicator matrix respectively. Note that we use the superscript to index the level in the tag taxonomy and the subscript to index the node-specific dictionary in that level.

Given an underlying tag taxonomy, we associate a separate dictionary with each tag node. These individual dictionaries are called node-specific dictionaries and they serve as local viewpoints for exploring the discriminative information among training regions from the same class or super-class. We concatenate the node-specific dictionaries in each level to construct a new large dictionary which is called a level-specific dictionary. Suppose that the level-specific dictionaries in the super-class and basic-class levels are learned and represented as $D^{(1)} = [D_1^{(1)}, D_2^{(1)}, ..., D_S^{(1)}] \in \mathbb{R}^{d\times J}$ and $D^{(2)} = [D_1^{(2)}, D_2^{(2)}, ..., D_K^{(2)}] \in \mathbb{R}^{d\times J}$, where $D_s^{(1)}$ and $D_k^{(2)}$ are associated with the s-th super-class and k-th class respectively. Given level-specific dictionaries $D^{(1)}, D^{(2)}$ and a region $\mathbf{x}_n \in \mathbb{R}^{d\times 1}$from the s-th superclass and k-th class, we obtain the group sparse representations $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ of this region as follows:

$$\begin{aligned}\mathbf{z}_n^{(1)} &= arg\min_{\mathbf{z}_n^{(1)}} \frac{1}{2}||\mathbf{x}_n - D^{(1)}\mathbf{z}_n^{(1)}||_2^2 + \lambda_1 \sum_{s=1}^{S} \beta_s^{(1)} ||z_{n_s}^{(1)}||_2 \\ \mathbf{z}_n^{(2)} &= arg\min_{\mathbf{z}_n^{(2)}} \frac{1}{2}||\mathbf{x}_n - D^{(2)}\mathbf{z}_n^{(2)}||_2^2 + \lambda_2 \sum_{k=1}^{K} \beta_k^{(2)} ||z_{n_k}^{(2)}||_2.\end{aligned} \tag{2}$$

Here we introduce $\mathbf{q}_n^{(1)}$ and $\mathbf{q}_n^{(2)}$ to denote the 'ideal' group sparse codes of $\mathbf{x}_n$ corresponding to $D^{(1)}$ and $D^{(2)}$ respectively. In particular, the non-zero values of $\mathbf{q}_n^{(1)}$ or $\mathbf{q}_n^{(2)}$ occur at those indices where the dictionary items belong to the node-specific dictionary $D_s^{(1)}$ or $D_k^{(2)}$. We use $Z^{(1)} = [\mathbf{z}_1^{(1)}, ..., \mathbf{z}_N^{(1)}] \in \mathbb{R}^{J\times N}$ to denote the group sparse codes of all regions at the super-class level. The matrices $Z^{(2)}, Q^{(1)}, Q^{(2)}$ are defined in a similar way.

Based on the sparse representations from the super-class and basic-class levels, we aim to learn two linear classifiers denoted as $f^{(1)}(\mathbf{z}, W_s) = W_s\mathbf{z}$ and $f^{(2)}(\mathbf{z}, W) = W\mathbf{z}$ for the two levels respectively, where $W_s \in \mathbb{R}^{S\times J}$ and $W \in \mathbb{R}^{K\times J}$. The objective function for learning all the dictionaries and classifiers are formulated as:

$$\begin{aligned}\min_{D^{(i)2}_{i=1}, W_s, W} & ||H^{(1)} - W_s Z^{(1)}||^2 + ||H^{(2)} - W(Z^{(1)} + Z^{(2)})||^2 \\ & + \nu(||Q^{(1)} - Z^{(1)}||^2 + ||Q^{(2)} - Z^{(2)}||^2) \\ & + \mu(||W_s||_2^2 + ||W||_2^2)\end{aligned} \tag{3}$$

where $Z^{(1)} = [\mathbf{z}_1^{(1)}, ..., \mathbf{z}_N^{(1)}], Z^{(2)} = [\mathbf{z}_1^{(2)}, ..., \mathbf{z}_N^{(2)}]$

$$Q^{(1)} = [\mathbf{q}_1^{(1)}, ..., \mathbf{q}_N^{(1)}], Q^{(2)} = [\mathbf{q}_1^{(2)}, ..., \mathbf{q}_N^{(2)}].$$

Note that this is a constrained optimization problem where the constraint is that matrices $Z^{(1)}$ and $Z^{(2)}$ are obtained by minimizing the reconstruction error with group LASSO penalty from the basic-class and super-class levels as shown in (2). This objective function consists of two parts:

1. The first part is the classification error from each level as shown in the first line of (3). The two classifiers W_s and W are learned by the linear regression. Note that

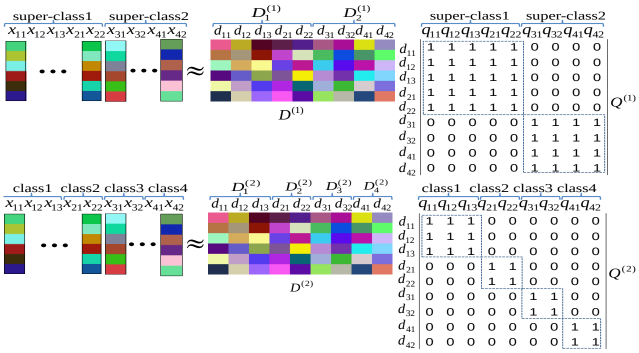


Figure 2. **An example of the *ideal* sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ for classification task.** Given nine image regions (on the leftmost) come from four basic-classes and two super-classes, we learn two level-specific dictionaries for the super-class and basic-class levels respectively. The super-class level dictionary is defined as: $D^{(1)} = [D_1^{(1)}, D_2^{(1)}]$ while the basic-level dictionary is $D^{(2)} = [D_1^{(2)}, D_2^{(2)}, D_3^{(2)}, D_4^{(2)}]$. For each region from one labeled tag, we aim to use only the node-specific dictionary that is associated with the same tag to reconstruct the region. This is because image regions from the same basic-class or super-class are more likely to share visual features and thus can be used to reconstruct each other.

W_s is not used for final region tagging. W_s is learned to guarantee that the sparse codes obtained from the super-class level are discriminative and thus can be used to help learn W for the basic-class level.

2. The second part is the regularization of sparse codes from two levels as shown in the second line of (3). The *ideal* sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ are block-diagonal as shown in Figure 2. We call sparse codes matrices $Q^{(1)}$ and $Q^{(2)}$ *ideal* because they are *ideal* for classification task. We minimize the difference between the true sparse codes and the corresponding *ideal* sparse codes to encourage the true sparse codes to be close to the *ideal* sparse codes. It means that for training regions X_k from the k-th class and X_s from the s-th super-class, we encourage the corresponding node-dictionaries $D_s^{(1)}$ and $D_k^{(2)}$ to be selected for group sparse coding. In addition, the non-zeros in $Q^{(2)}$ are a subset of non-zeros in $Q^{(1)}$. Note that this fixed and structured relationship between $Q^{(1)}$ and $Q^{(2)}$ regularizes the relationship between $Z^{(1)}$ and $Z^{(2)}$ from two levels, which makes it possible to use sparse codes from different levels to improve classification accuracy.

Note that we use the sum of sparse codes from two levels as the features to design the class classifier W for two reasons. First, we make use of the discriminative information encoded in the sparse codes obtained from the super-class level to learn W. Second, it encourage classes within the same super-class to implicitly share sparse codes obtained from super-class level. This can handle the situation where the training classes are very unbalanced. For example, there are many training regions for the tag *cat* but little training regions for *dog*. Given the feature of an image region from *dog*, it can be reconstructed using the level-specific dictionary from the basic-class level, which may activate multiple node-specific dictionaries in the basic-class level. This is due to the little training regions for the tag *dog* and it will be difficult to classify the class label of this image region. However, when using the level-specific dictionary from the super-class level to reconstruct this image region, it may only activate the node-specific dictionary associated with the super-class *animal*. This is because other tags within the same super-class *animal* may share some features with *dog* and can help to represent this image region better other than *dog* itself. Even if we cannot classify this image region as *dog*, we can at least classify this image regions as other tags that belong to the super-class *animal* instead of totally uncorrelated tags from other super-classes. Thus using the sum of sparse codes from two levels as features for designing the class classifiers can support this implicit feature sharing among classes within the same super-class.

3.3. Optimization Algorithm

Motivated by [12], we propose a stochastic gradient descent algorithm for optimizing the objective function. We first rewrite the objective function in (3) as follows:

$$\min_{D^{(i)}{}_{i=1}^{2}, W_s, W} \sum_{i=1}^{N} \ell^n(D^{(1)}, D^{(2)}, W_s, W) + \mu(||W_s||_2^2 + ||W||_2^2)$$

where

$$\begin{aligned} \ell^n &= \nu(||\mathbf{q}_n^{(1)} - \mathbf{z}_n^{(1)}||^2 + ||\mathbf{q}_n^{(2)} - \mathbf{z}_n^{(2)}||^2) \\ &\quad + ||\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)}||^2 + ||\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})||^2. \end{aligned} \tag{4}$$

Note that the sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ are functions of $D^{(1)}$ and $D^{(2)}$ respectively. We use the notation $\ell^n(D^{(1)}, D^{(2)}, W_s, W)$ to emphasize that the loss function associated with the n-th region is also a function of $D^{(1)}$ and $D^{(2)}$. We use the following procedure to optimize the objective function: first, we randomly select a training instance $(\mathbf{x}_n, \mathbf{h}_n^{(1)}, \mathbf{h}_n^{(2)})$ for the t-th iteration; next, we compute the sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ using $D^{(1)}$ and $D^{(2)}$ by (2); finally, we update $D^{(1)}, D^{(2)}, W_s$ and W by the gradients of the loss function ℓ^n with respect to them.

We next describe the methods for computing the gradients of the loss function ℓ^n with respect to the level-specific classifiers and dictionaries. When the sparse codes $z_n^{(1)}$ and $z_n^{(2)}$ are known, we can compute the gradient of ℓ^n with respect to W_s and W as follows:

$$\begin{aligned} \frac{\partial \ell^n}{\partial W_s} &= -2(\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)})\mathbf{z}_n^{(1)T} \\ \frac{\partial \ell^n}{\partial W} &= -2(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)}))(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})^T. \end{aligned} \tag{5}$$

We use the chain rule to compute the gradient of ℓ^n with respect to $D^{(1)}$ and $D^{(2)}$ as follows:

$$\frac{\partial \ell^n}{\partial D^{(1)}} = \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(1)}} \frac{\partial \mathbf{z}_n^{(1)}}{\partial D^{(1)}}, \frac{\partial \ell^n}{\partial D^{(2)}} = \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(2)}} \frac{\partial \mathbf{z}_n^{(2)}}{\partial D^{(2)}} \tag{6}$$

where

$$\begin{aligned} \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(1)}} =& -2W_s^T(\mathbf{h}_n^{(1)} - W_s\mathbf{z}_n^{(1)}) \\ & -2W^T(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})) - 2\nu(\mathbf{q}_n^{(1)} - \mathbf{z}_n^{(1)}) \\ \frac{\partial \ell^n}{\partial \mathbf{z}_n^{(2)}} =& -2W^T(\mathbf{h}_n^{(2)} - W(\mathbf{z}_n^{(1)} + \mathbf{z}_n^{(2)})) - 2\nu(\mathbf{q}_n^{(2)} - \mathbf{z}_n^{(2)}). \end{aligned}$$

To compute the gradient of $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ with respect to $D^{(1)}$ and $D^{(2)}$, we use implicit differentiation on the fixed point equation similar to [12, 24, 23]. We first establish the fixed point equation of (2) by calculating the derivatives of $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ on both sides and have:

$$\begin{aligned} D_\Lambda^{(1)T}(\mathbf{x}_n - D_\Lambda^{(1)}\mathbf{z}_{n\Lambda}^{(1)}) = \lambda_1\Gamma^{(1)}[\frac{z_{n_1\Lambda}^{(1)T}}{||z_{n_1\Lambda}^{(1)}||_2}, ..., \frac{z_{n_S\Lambda}^{(1)T}}{||z_{n_S\Lambda}^{(1)}||_2}]^T \\ D_\Lambda^{(2)T}(\mathbf{x}_n - D_\Lambda^{(2)}\mathbf{z}_{n\Lambda}^{(2)}) = \lambda_2\Gamma^{(2)}[\frac{z_{n_1\Lambda}^{(2)T}}{||z_{n_1\Lambda}^{(2}||_2}, ..., \frac{z_{n_K\Lambda}^{(2)T}}{||z_{n_K\Lambda}^{(2)}||_2}]^T \end{aligned} \tag{7}$$

where Λ denote the index set of non-zero sparse coefficients in $z_n^{(1)}$ and $z_n^{(2)}$. Both $\Gamma^{(1)}$ and $\Gamma^{(2)}$ are block-diagonal. The s-th block in $\Gamma^{(1)}$ is $\beta_s^{(1)} I_s$ while the k-th block in $\Gamma^{(2)}$ is $\beta_k^{(2)} I_k$, where I_s, I_k are the corresponding identity matrices. We calculate the derivatives of $D^{(1)}$ and $D^{(2)}$ on both sides of (7), and have

$$\begin{aligned} \frac{\partial \mathbf{z}_{n\Lambda}^{(1)}}{\partial D_\Lambda^{(1)}} = (D_\Lambda^{(1)T} D_\Lambda^{(1)} + \lambda_1\Gamma^{(1)}A^{(1)})^{-1}[\frac{\partial D_\Lambda^{(1)T}\mathbf{x}_n}{\partial D_\Lambda^{(1)}} - \frac{\partial D_\Lambda^{(1)T} D_\Lambda^{(1)}}{\partial D_\Lambda^{(1)}}\mathbf{z}_{n\Lambda}^{(1)}] \\ \frac{\partial \mathbf{z}_{n\Lambda}^{(2)}}{\partial D_\Lambda^{(2)}} = (D_\Lambda^{(2)T} D_\Lambda^{(2)} + \lambda_2\Gamma^{(2)}A^{(2)})^{-1}[\frac{\partial D_\Lambda^{(2)T}\mathbf{x}_n}{\partial D_\Lambda^{(2)}} - \frac{\partial D_\Lambda^{(2)T} D_\Lambda^{(2)}}{\partial D_\Lambda^{(2)}}\mathbf{z}_{n\Lambda}^{(2)}] \end{aligned}$$

where the matrices $A^{(1)}$ and $A^{(2)}$ are block-diagonal and the s-th block in $A^{(1)}$ is $\frac{||z_{n_s\Lambda}^{(1)}||I_s - z_{n_s\Lambda}^{(1)} z_{n_s\Lambda}^{(1)T}}{||z_{n_s\Lambda}^{(1)}||_2^2}$ while the k-th block in $A^{(2)}$ is $\frac{||z_{n_k\Lambda}^{(2)}||I_k - z_{n_k\Lambda}^{(2)} z_{n_k\Lambda}^{(2)T}}{||z_{n_k\Lambda}^{(2)}||_2^2}$. Therefore, (6) can be rewritten as

$$\begin{aligned} \frac{\partial \ell^n}{\partial D^{(1)}} &= -D^{(1)}\mathbf{s}_n^{(1)}\mathbf{z}_n^{(1)T} + (\mathbf{x}_n - D^{(1)}\mathbf{z}_n^{(1)})\mathbf{s}_n^{(1)T} \\ \frac{\partial \ell^n}{\partial D^{(2)}} &= -D^{(2)}\mathbf{s}_n^{(2)}\mathbf{z}_n^{(2)T} + (\mathbf{x}_n - D^{(2)}\mathbf{z}_n^{(2)})\mathbf{s}_n^{(2)T} \end{aligned} \tag{8}$$

where the auxiliary variables $\mathbf{s}_n^{(1)}$ and $\mathbf{s}_n^{(2)}$ are defined as follows:

$$\mathbf{s}_{\Lambda^C}^{(1)} = 0, \mathbf{s}_\Lambda^{(1)} = (D_\Lambda^{(1)T} D_\Lambda^{(1)} + \lambda_1\Gamma^{(1)}A^{(1)})^{-1}\frac{\partial \ell^n}{\partial \mathbf{z}_{n\Lambda}^{(1)}}$$

$$\mathbf{s}_{\Lambda^C}^{(2)} = 0, \mathbf{s}_\Lambda^{(2)} = (D_\Lambda^{(2)T} D_\Lambda^{(2)} + \lambda_2\Gamma^{(2)}A^{(2)})^{-1}\frac{\partial \ell^n}{\partial \mathbf{z}_{n\Lambda}^{(2)}}.$$

The steps $1 - 15$ in Algorithm 1 summarize our joint learning algorithm.

Algorithm 1 Multi-layer Supervised Dictionary Learning for Region Tagging (MSDL)

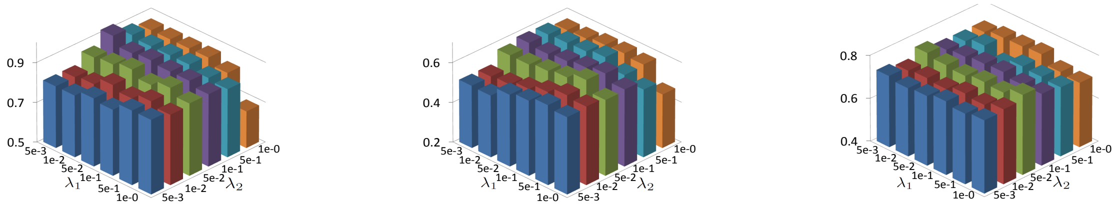
```
1:  **Part 1: Dictionary Learning**
2:  **Input:** $X$ (training regions), $H^{(1)}$ (super-class label indicator matrix), $H^{(2)}$
    (class label indicator matrix), **D** (initial dictionary), $T$ (number of iterations), $N$
    (number of training samples), $\rho$ (initial learning rate), $\nu$, $\mu$, $n_0 = \frac{T}{10}$.
3:  **Output:** classifiers $W_s$ and $W$; dictionaries $D^{(1)}$ and $D^{(2)}$
4:  **for** $t = 1 ... T$ **do**
5:      Permute training samples $(X, H^{(1)}, H^{(2)})$;
6:      **for** $n = 1 ... N$ **do**
7:          Evaluate the group sparse codes $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_n^{(2)}$ of the region $\mathbf{x}_n$;
8:          Choose the learning rate $\rho_t = \min(\rho, \rho * n_0/n)$
9:          Update the classifiers and dictionaries by a projected gradient step
10:         $W_s \leftarrow \prod_{W_s}[W_s - \rho_t(\frac{\partial \ell^n}{\partial W_s} + \mu W_s)]$;
11:         $W \leftarrow \prod_W[W - \rho_t(\frac{\partial \ell^n}{\partial W} + \mu W)]$;
12:         $D^{(1)} \leftarrow \prod_{D^{(1)}}[D^{(1)} - \rho_t \frac{\partial \ell^n}{\partial D^{(1)}}]$
13:         $D^{(2)} \leftarrow \prod_{D^{(2)}}[D^{(2)} - \rho_t \frac{\partial \ell^n}{\partial D^{(2)}}]$
14:     **end for**
15: **end for**
16: **Part 2: Region Tagging**
17: **Input:** $\hat{\mathbf{x}}$ (test region)
18: **Output:** $\hat{y}$ (predicted tag class)
19: Evaluate the group sparse codes $\hat{\mathbf{z}}^{(1)}$ and $\hat{\mathbf{z}}^{(2)}$ of the test region $\hat{\mathbf{x}}$;
20: The predicted tag for this test region is $\hat{y} = arg\max_j W(\hat{\mathbf{z}}^{(1)} + \hat{\mathbf{z}}^{(2)})$.
```

4. Experiments

4.1. Datasets and Feature Extraction

We evaluated our approach for region tagging using several benchmarks, including MSRC-v1, MSRC-v2 [19], and SAIAPR TC-12 datasets [2]. Images in these datasets have been segmented into regions and their ground truth of region masks are also provided. MSRC-v1 contains 240 images that are segmented into 562 regions associated with 13 tags, whereas MSRC-v2 has 591 images and 1482 regions associated with 23 tags. And SAIAPR TC-12 contains 99,535 regions segmented from 20,000 images. The associated 276 tags for this dataset are organized into a hierarchy.

We follow the protocol in [7] to extract RGB color features and sample training and test regions. We use 8 bins for each color channel and count the ratio of pixels whose RGB values fall into each bin to construct a $3D$ histogram. Thus each image region is represented as a 512-dimensional RGB color histogram. For the MSRC-v1 dataset, we randomly sample 200 images and the corresponding regions as the training set, whereas for the MSRC-v2 dataset, 471 images are randomly sampled to form the training set. The remaining regions are used for testing. For SAIAPR TC-12 dataset, we select the same 27 localized tags out of 276 tags as in [7] for evaluation. Then we randomly select 2500 regions whose tags are within the selected subset of 27 tags as the training set and another 500 regions as the test set.



(a) MSRC-v1 (b) MSRC-v2 (c) SAIAPR TC-12

Figure 3. **The effect of parameters λ_1 and λ_2 on the region tagging performance of our method on three datasets.**

Methods	MSRC-v1	MSRC-v2	SAIAPR
Lasso[20]	0.612	0.448	0.652
Group Lasso[27]	0.636	0.458	0.598
Sparse Group Lasso[4]	0.625	0.433	0.561
SGSC[24]	0.726	0.460	-
G^{2}SRRT(kNN)[7]	0.727	0.473	0.646
G^{2}SRRT(kNN+Tag)[7]	0.739	0.533	0.667
SSDL	**0.830**	**0.560**	**0.704**
MSDL	**0.926**	**0.634**	**0.772**

Table 1. **The average accuracies of region tagging by different methods on MSRC-v1, MSRC-v2 and SAIAPR TC-12 datatsets.**

4.2. Comparing Methods and Parameter Setting

As in [25, 7], we choose LASSO [20], Group LASSO [27] and Sparse Group LASSO [4] as baselines and use the implementation of these methods in SLEP package [10]. We compare our mutli-layer supervised dictionary learning method (MSDL) with two state-of-the-art approaches: SGSC [25], G^{2}SRRT [7]. In order to demonstrate that the super-class level can help improve the accuracy of region tagging, we use single-layer supervised dictionary learning (SSDL) corresponding to the basic-class level as another baseline. The performance of tagging accuracy (number of correctly classified regions over the total test regions) is reported as the average over 5 different trials corresponding to different partitions of training and test sets.

There are two important parameters in our model: λ_1 and λ_2 that are used to balance the reconstruction error and the sparse penalty for two levels. The ranges of both λ_1 and λ_2 for all datasets are $\{0.005,0.01,0.05,0.1,0.5,1\}$. For other parameters in all experiments, we set the parameters $\nu = 0.1$ and $\mu = 0.001$ for the regularization of sparse codes and classifiers respectively. In addition, the initial learning rate ρ is set to be 0.001 and the level-specific dictionaries are initialized using the software SPAMS [13]. The performance of region tagging by our method with different λ_1 and λ_2 on three datasets are illustrated in Figure 3. We see that the highest performance is achieved at different values of the two parameters for the three datasets.

4.3. Experimental Results

The accuracies of region tagging using different methods on three datasets are summarized in Table 1. We can see that for all the datasets, both SSDL and our method outperform all the other methods. In particular, when compared with other sparse coding-based algorithms, SSDL and our method significantly improve the performance for region tagging on MSRC-v1 dataset—by a margin close to 10% and 20% respectively. This is because the labeled tag distribution in MSRC-v1 is very unbalanced and the tag with most training regions is more likely to be selected for reconstruction of test regions when using the group sparse coding algorithm. On the contrary, both SSDL and our method can reduce the reconstruction error to some extent by learning a more reconstructive and discriminative dictionary. Furthermore, for the MSRC-v2 and SAIAPR TC-12 datasets, our method improves the tagging accuracy by 10% that is twice than the improvement obtained by SSDL. And this good performance by our method demonstrates that, we effectively explored the semantic relationship among tags and make the super-class level help improve the performance for region tagging. In addition, different from the MSRC datasets, images in the SAIAPR TC-12 dataset are more arbitrary and image regions from the same tag vary drastically; the better performance by our method further demonstrates that our approach can handle the diversity and arbitrariness of image content by exploiting hierarchial relationships among tags. Finally, note that the algorithm SGSC [25] needs to build a spatial kernel for regions within each image, which requires regions within each image to be jointly selected and included in the training and test sets. Since we randomly sampled image regions of the SAIAPR TC-12 dataset and the spatial kernel might not be built, the performance for region tagging by SGSC is not reported in Table 1 as in [7].

Figures 4 and 6 illustrate two tag taxonomies associated with MSRC-v1 and MSRC-v2 respectively while Figures 5 and 7 display the corresponding confusion matrices obtained by SSDL and our method under the two datasets. Since we obtain similar results in MSRCv1 and MSRCv2 datasets, for simplicity we take MSRC-v1 dataset for analysis. Comparing the confusion matrix obtained by SSDL with our method in Figure 5, we can see that tags *building*, *tree*, *cow*, *aeroplane*, *bicycle* have large improvements in tagging accuracy using our proposed method. Moreover, instead of classifying regions from the tag *horse* as *face* by SSDL, our method classifies them as *cow* which is also in the same super-class as *horse*. This demonstrates how our method takes advantages of implicit sharing of sparse codes

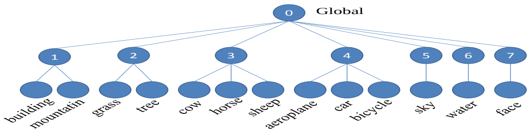


Figure 4. **The tag taxonomy for MSRC-v1**

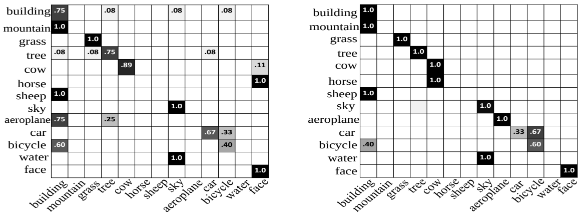


Figure 5. **Confusion matrices for SSDL (left) and our method MSDL (right) on the MSRC-v1 dataset.**

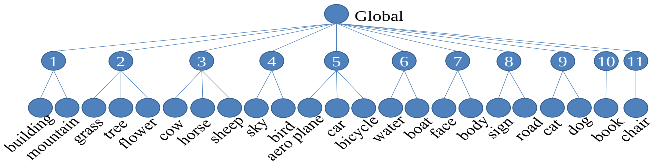


Figure 6. **The tag taxonomy for MSRC-v2**

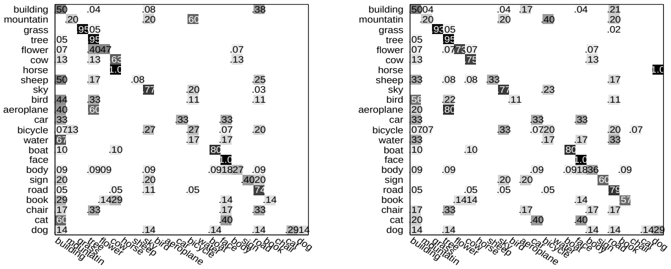


Figure 7. **Confusion matrices for SSDL (left) and our method MSDL (right) on the MSRC-v2 dataset.**

obtained from the super-class level to help improve the accuracy of tag nodes from the basic-class level. It is also interesting to note that the tag *car* has a slight decrease in tagging accuracy because some regions from *car* are misclassified as *bicycle* which is also in the same-super class. Thus, different tags benefit in different degrees from the implicit sharing of sparse codes and a similar phenomenon has also been observed in [18] which uses a parameter sharing strategy.

To further investigate the performance of region tagging by SSDL and our method, we select nine tags in each dataset and report the corresponding tagging accuracy of each tag in Figure 8. From the detailed tagging performance, we can see that our method obtains better tagging performance for most of the tags. However, it is also interesting to note that SSDL obtains a slightly better performance for some tags such as *car* in MSRC-v1 dataset and *water* in SAIAPR TC-12 dataset. One possible reason is that the visual appearances of image regions from these tags are very different from other tags within the same superclass which introduces a negative transfer. Similar facts are also observed in [18].

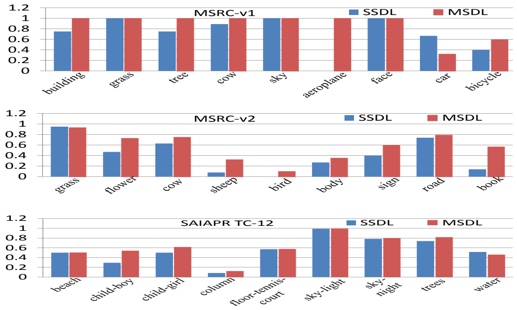


Figure 8. **The performance comparison using SSDL and MSDL for nine selected tags on each dataset.**

Figure 9 shows some examples of region tagging results on three datasets. We see that our method correctly classifies those regions that are misclassified by [7] and SSDL.

5. Conclusion

In this paper, we have proposed a multi-layer hierarchical supervised dictionary learning framework for region tagging by exploring the given tag taxonomy. Specifically, we associate each tag node in the taxonomy with one node-specific dictionary and concatenate the node-specific dictionaries in each level to construct a level-specific dictionary. Using the level-specific dictionary and corresponding level-specific group structure, we obtain level-specific sparse codes that are also close to the *ideal* sparse codes. The sparse codes from different levels are summed up as the final feature representation to learn the level-specific classifier. This enables us to simultaneously take advantages of the robust encoding ability of group sparse coding as well as the semantic relationship in the tag taxonomy. We have extensively tested our approach on three benchmark datasets and results clearly confirm the effectiveness of our approach for region tagging. Although in this paper we select region tagging to evaluate our proposed method, we believe that it is a general method and can be developed and applied to object and activity recognition.

Acknowledgement

This work was supported by an ONR grant N00014-12-1-0124.

References

[1] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse learning. In *UAI*, 2011. 2

[2] H. J. Escalante, C. A. Hernández, J. A. González, A. López-López, M. M. y Gómez, E. F. Morales, L. E. Sucar, L. V. Pineda, and M. Grubinger. The segmented and annotated iapr tc-12 benchmark. *Computer Vision and Image Understanding*, 2010. 5

[3] J. Fan, Y. Gao, and H. Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *ACM Multimedia*, 2004. 1, 2

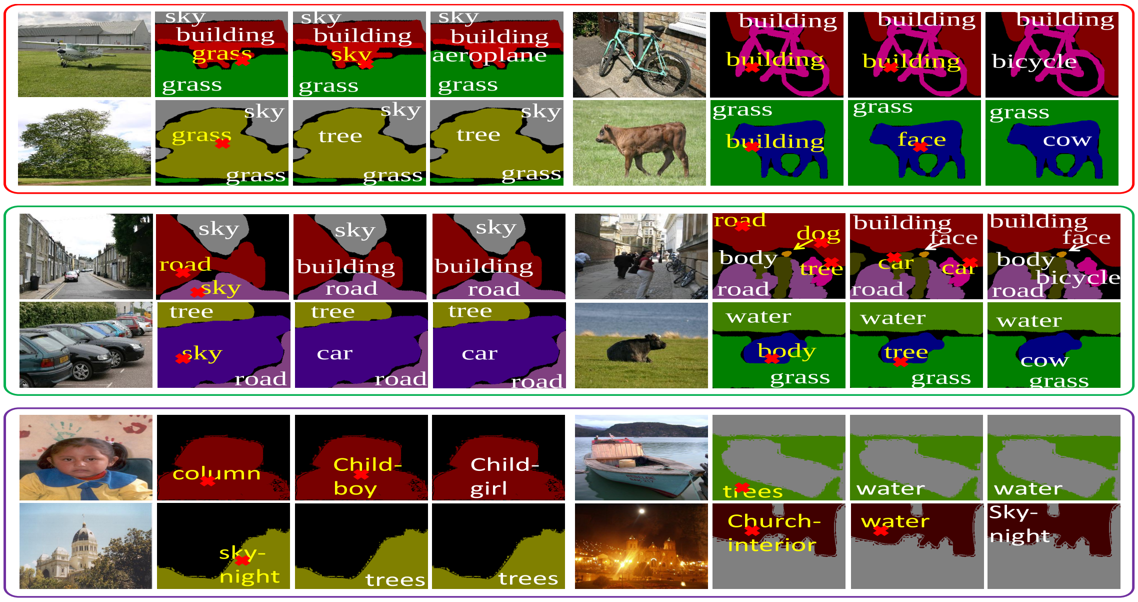


Figure 9. **Examples of region tagging results on three benchmark image datasets.** The subfigures from the top to the bottom corresponds to the MSRC-v1, MSRC-v2 and SAIAPR TC-12 datasets respectively. In each subfigure, the columns from the left to the right correspond to the samples image, region tagging results by [7], our baseline (SSDL) and our method (MSDL). Misclassified tags are in *yellow* while correctly classified tags are in white. The figure is best viewed in color.

[4] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. Technical Report arXiv, 2010. 6

[5] S. Gao, L.-T. Chia, and I. W.-H. Tsang. Multi-layer group sparse coding - for concurrent image classification and annotation. In *CVPR*, 2011. 2

[6] G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *CVPR*, 2008. 1

[7] Y. Han, F. Wu, J. Shao, Q. Tian, and Y. Zhuang. Graph-guided sparse reconstruction for region tagging. In *CVPR*, 2012. 1, 2, 5, 6, 7, 8

[8] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 2011. 2

[9] D. Liu, S. Yan, Y. Rui, and H.-J. Zhang. Unified tag analysis with multi-edge graph. In *ACM Multimedia*, 2010. 2

[10] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009. 6

[11] X. Liu, B. Cheng, S. Yan, J. Tang, T.-S. Chua, and H. Jin. Label to region by bi-layer sparsity priors. In *ACM Multimedia*, 2009. 1, 2

[12] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012. 4, 5

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, 2009. 6

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2008. 2

[15] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *CVPR*, 2007. 1

[16] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *ECCV*, 2008. 1

[17] D.-S. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *CVPR*, 2008. 2

[18] R. Salakhutdinov, A. Torralba, and J. B. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011. 7

[19] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 2009. 5

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1994. 3, 6

[21] C. Yang, M. Dong, and F. Fotouhi. Region based image annotation through multiple-instance learning. In *ACM Multimedia*, 2005. 1, 2

[22] C. Yang, M. Dong, and J. Hua. Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In *CVPR*, 2006. 1, 2

[23] J. Yang and M.-H. Yang. Top-down visual saliency via joint crf and dictionary learning. In *CVPR*, 2012. 5

[24] J. Yang, K. Yu, and T. S. Huang. Supervised translation-invariant sparse coding. In *CVPR*, 2010. 2, 5, 6

[25] Y. Yang, Y. Yang, Z. Huang, H. T. Shen, and F. Nie. Tag localization with spatial correlations and joint group sparsity. In *CVPR*, 2011. 1, 2, 6

[26] J. Yuan, J. Li, and B. Zhang. Exploiting spatial context constraints for automatic image region annotation. In *ACM Multimedia*, 2007. 1, 2

[27] M. Yuan, M. Yuan, Y. Lin, and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 2006. 2, 3, 6

[28] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, 2010. 2