Local image tagging via graph regularized joint group sparsity

Yang Yang a, Zi Huang a, Yi Yang b, Jiajun Liu a, Heng Tao Shen a,*, Jiebo Luo c

a School of Information Technology and Electrical Engineering, The University of Queensland, Australia
b School of Computer Science, Carnegie Mellon University, USA
c Department of Computer Science, University of Rochester, USA

A R T I C L E   I N F O

Article info

Received 30 April 2012
Received in revised form 30 September 2012
Accepted 18 October 2012
Available online 20 November 2012

Keywords:
Local image tagging
Group sparse coding
Graph regularization
Tag propagation

A B S T R A C T

In recent years, massive amounts of web image data have been emerging on the web. How to precisely label these images is critical and challenging to modern image search engines. Due to the fact that web image contents are more and more complex, existing image-level tagging methods may become less effective and hardly achieve satisfactory performance. This raises an urgent need for the fine-grained tagging, e.g., region-level tagging. In this work, we study how to establish mapping between tags and image regions. In particular, a novel hierarchical local image tagging method is proposed to simultaneously assign tags to all the regions within the same image. We propose a Laplacian Joint Group Lasso (LJGL) model to jointly reconstruct the regions within a test image with a set of labeled training data. The LJGL model not only considers the robust encoding ability of joint group lasso but also preserves local structural information embedded in test regions. Besides, we extend the LJGL model to a kernel version in order to achieve the non-linear reconstruction. An effective algorithm is devised to optimize the objective function of the proposed model. Tags of training data are propagated to the reconstructed regions according to the reconstruction coefficients. Extensive experiments on four public image datasets demonstrate that our proposed models achieve significant performance improvements over the state-of-the-art methods in local image tagging.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Multimedia content analysis is a fundamental research problem in computer vision and multimedia research areas [6,24,32,38]. During the past few years, confronted with the explosive growth of user-generated images on the web, existing image-level tagging methods become less and less effective due to the high complexity of web image contents. How to precisely understand and tag images at fine-grained levels becomes a great challenge to modern image search engines. In this paper, we aim to address the problem of tagging images at the region level, i.e., assigning tags to image regions.

During the last decade, several endeavors have been made towards this research topic. Multiple instance learning [8] has been applied to region based image annotation [34,35] and shown their effectiveness to some extent. These approaches employed image regions in both the training and test phases, but they actually focused on image-level annotation. Most recently, sparse coding has been widely applied to this problem. Liu et al. [13,14] proposed the Bi-Layer sparse coding for encoding image regions and propagating tags at the region level. In their works, images were first segmented into basic regions, then the Bi-Layer model was applied to reconstruct each test region from a dictionary formed by other basic regions. The common tags of images containing the target region and sparsely selected regions will be re-assigned to the target region according to the reconstruction coefficients. It is worth noting that basic regions in the dictionary are implicitly assumed to be independent from each other. Contextual relationships among these semantic regions/objects, e.g., co-occurrence and spatial correlations, are ignored. Besides, when reconstructing regions within an image they individually encode each region and again ignore the intrinsic correlations among encoding regions. All of these correlations are important clues for uncovering the underlying data structure, and neglecting them may lead to a potential loss in interpretability and reconstruction performance. Yang et al. [41] proposed a joint group lasso which was able to reconstruct all the test regions at the same time. It considered the implicit correlations among test regions and the spatial correlations among basic regions. However, both of the above works are based on the assumption that test regions can be linearly reconstructed, which is not always the case in reality, and also the intrinsic data structural information among test regions is not well preserved.

In general, an image contains a limited number of objects, which indicates that the image can only be related to a limited number of samples in a large enough image database. When the
image and/or its regions are reconstructed, it would be beneficial to utilize the limited number of related samples rather than those unrelated ones which would probably introduce potential noise. Therefore, a reasonable assumption is that the number of training samples that are identified by the propose model and used to reconstruct the test samples should be small compared to the whole number of samples in the database. Based on the above observations and intuitions, in this study we present an efficient and context-aware local image tagging approach, which jointly assigns tags to image regions within a test image. Given a test image, we first search for its potentially relevant nearest neighbors in the database to form a reconstruction dictionary. The test image is then segmented into several regions, which are subsequently reconstructed based on the dictionary rather than all the images in the dataset. Besides, the proposed approach can also prevent introducing unnecessary noise in the tag propagation phase. The reconstruction task is achieved by a Laplacian Joint Group Lasso (LJGL) model, which integrates a Laplacian prior in order to preserve the local data structure among the reconstructed regions. Furthermore, we also extend our proposed model to a kernel version so as to perform the non-linear reconstruction. We summarize the overall framework of our local image tagging approaches in Fig. 1.

The contributions of this paper are summarized as follows:

1. A novel hierarchical local image tagging method is developed for setting up mappings between tags and image regions. Given a test image, instead of all other images in the dataset, its $k$ nearest neighbors are first retrieved to construct the dictionary. This helps avoid noises in tag propagation and boost the tagging efficiency.

2. A Laplacian Joint Group Lasso (LJGL) is proposed for joint region reconstruction. The LJGL not only takes the robust encoding ability of joint group lasso into account but also preserves the local data structural information among the test regions.

3. A kernel version of the LJGL model is extended to cope with the non-linear reconstruction cases.

4. We devise a novel effective algorithm for optimizing the objective function of the LJGL. Theoretical proof and analysis are given to guarantee that the algorithm converges to the global optimum.

5. Extensive experiments are conducted on four public image datasets to show the effectiveness of our tagging method.

The rest of this paper is organized as follows. Related work will be reviewed in Section 2, followed by describing the details of the proposed LJGL model, its non-linear kernel version and an algorithm for optimization in Section 3. Section 4 reports all the experimental results, followed by the conclusion in Section 5.

2. Related work

In this section, we provide a brief review on image tagging/annotation, including image-level and region-based approaches. Besides, we also briefly review related work on sparse coding.

Image-level tagging refers to the methods that assign tags to the whole image. Wang et al. [28] proposed to identify a group of potential tags from surrounding text by summarizing tags’ tf-idf weights first, and then these potential tags were refined with a random walk with restart model. In [11], Li et al. proposed a neighbor voting algorithm which can establish tag relevance with respect to images by accumulating votes from their visually similar neighbors. Wang et al. [32] first searched visually and semantically similar images, then mined the results by a clustering algorithm to identify latent terms which are treated as the final tags.

Whereas tagging is designed to simulate general users’ tagging behavior and usually solved by data-driven approaches [37,46], the tags used in image tagging usually have no explicit limitations. Multimedia annotation is normally formulated as a classification problem. The available annotations are limited in a small lexicon. In [17], Moxley et al. proposed a graph reinforcement model by extending semi-supervised learning to perform tag propagation task. They first constructed several nearest neighbor graphs by searching visually similar videos in multiple modalities, and then utilized the proposed graph reinforcement mining approach to re-rank the annotations. In [20], Qi et al. formulated video annotation as a multi-label classification problem and proposed a Correlative Multi-Label framework to simultaneously model multi-concepts and take into account semantic correlations between concepts. They also illustrate that their model can be intuitively interpreted by Gibbs Random Field. By extending this idea, Hua et al. [9] further proposed a novel multi-label active learning framework which adopts an online learner instead of SVM-like classifier to deal with large-scale data. Wu et al. [33] focused on learning an optimal similarity metric by exploiting side information associated with social media, such as surrounding text and existing tags. After that, they used nearest neighbors classifier to identify tags for test images. Similarly, Mei et al. [16] also defined and learned a semantic distance function to measure similarity containing more semantic information. Recently, tags in images have also been transferred for effective video tagging [40].
Region-based image tagging/annotation has been proposed to perform the tagging task at more fine-grained levels, and a popular trend in this research topic is to employ supervised learning models to formulate region-based annotation as a classification problem. Singhal et al. [25] proposed to adapt the existing material detectors with spatial contextual information to enhance scene content understanding. The work in [5] utilized a conditional random field and integrate label co-occurrence and spatial context to identify the optimal label arrangement for several image regions. Saathoff et al. [21] proposed to learn a statistical classifier for each individual concept based on wavelet features and the spatial arrangement among concepts. A subsequent postprocessing step was added to refine the initial annotations. As aforementioned, the work in [13] proposed a Bi-Layer sparse coding to encode test regions for transferring tags. Several region-based image/video annotation methods are built based on multiple instance learning [8,34,35]. Nevertheless, although these methods exploit regions in the training phase, their objectives were to assign annotations to the whole image rather than each individual region. It is worth noting that our work is also related to semantic image decomposition, which parses an image into a set of components. Li et al. [10] proposed an image decomposition approach, which considers the contextual information of labels across images and identifies a local representation for each individual label. Different from this work, our work focuses on exploring correlations among image regions to predict the presence of labels for local image regions, and the above work applied their image decomposition approach to assign labels to the whole image.

Finally, we would also like to review the development of sparse coding which has been a fairly popular technique in the recent computer vision research. Yang et al. [36] improved vector quantization by extending sparse coding with spatial order of local descriptors. Gao et al. [6] followed this work and posed an additional constraint for enforcing maximal similarity preservation among similar descriptors. Mairal et al. [15] proposed simultaneous sparse coding to encode a group of similar patches for image restoration. Similarly in [2], Bengio et al. proposed a variant of sparse coding for jointly encoding the group of visual descriptors within the same image to achieve image-level sparsity. Wang et al. [29] proposed to use sparse coding twice for image annotation. They applied sparse coding to reconstruct images for establishing relations among images. The coefficients were used for dimensionality reduction over the feature representations. In [45], the authors applied group sparse coding to perform feature selection for image annotation. Han et al. [4] also proposed to use a structural group sparsity for feature selection and boost annotation performance by exploiting correlations among multiple tags. In [43,44], similar ideas to our model were proposed. They applied a multi-task joint sparse coding to image classification on multiple modalities. However, they overlooked the correlations among the basis in the training dictionary, which may provide more reconstructive ability.

Our work is also developed from sparse coding but different from Liu’s [13] and Yang’s [41] works. Compared with them, our model not only considers the correlations amongst the training regions in the dictionary but also explicitly preserves the local data structural information among the test regions. Also the kernel version of the LJGL can handle the non-linear reconstruction cases.

3. Laplacian joint group lasso

In this section, we propose a local image tagging approach by uncovering how a group of test regions can be jointly encoded from a set of potentially relevant images. We first introduce some preliminaries on sparse coding, including group lasso and joint group lasso. Then we introduce the LJGL model and its kernel variation by integrating a Laplacian prior to preserve the local structural information. Finally we design an effective algorithm for globally optimizing the objective function of the LJGL with theoretical guarantee.

3.1. Preliminary on sparse coding

3.1.1. Sparse coding

Sparse representation has shown its effectiveness in computer vision due to the computational benefits and robustness. It assumes that a signal \( y \in \mathbb{R}^N \) can be encoded by the sparse linear combination of \( N \) basic elements

\[
\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_p
\]

(1)

where \( X \in \mathbb{R}^{d \times N} \) is the encoding dictionary, \( \beta \) indicates the encoding coefficients, \( \lambda \) is a trade-off parameter and \( \| \cdot \|_p \) is the \( \ell_p \)-norm. Ideally, the “pseudo-norm” \( \ell_0 \)-norm can guarantee to obtain the sparsest solution, but it has been proven to be an NP-hard selection problem. In practice one usually instead uses the \( \ell_1 \)-norm to reformulate sparse coding as a convex problem, which is known as the Lasso [27].

3.1.2. Group sparse coding

Although the Lasso enjoys significant computational strength and great performance, it is worth noting that this method implicitly assumes that an element in the dictionary is independent of all others. In image region encoding, if we simply concatenate regions of the training images to form the dictionary [13], apparently we lose correlation clues among the regions within the same image, such as co-occurrence between objects (e.g., an image depicting a computer screen probably contains a computer keyboard), spatial dependency amongst regions (e.g., sky often lays over boats), etc. Besides, the Lasso tends to select more images because the \( \ell_1 \)-norm only guarantees region-level sparsity rather than image-level sparsity. This may introduce more potential noises when tag propagation is performed. Therefore, our first motivation is to integrate correlations among training regions and realize image-level sparsity.

Given an image dictionary \( X = [X_1, X_2, \ldots, X_G] \) consisting of \( G \) images, where the \( g \)th image is segmented into a group of \( N_g \) regions \( X_g = [X_{g1}, \ldots, X_{gN_g}] \) segmented from the \( g \)th image. To achieve the integration of correlations among the training regions, Group Lasso [42] can be applied to reformulate the reconstruction process for a test region \( y \in \mathbb{R}^d \)

\[
\min_{\beta} \frac{1}{2} \|y - \sum_{g=1}^G X_g \beta_g\|_2^2 + \lambda \sum_{g=1}^G \|\beta_g\|_2
\]

(2)

where \( \beta_g \in \mathbb{R}^{N_g}, g = 1, 2, \ldots, G \) are the encoding coefficients corresponding to the images in the dictionary. The Group Lasso uses a group-sparsity-inducing regularization instead of the \( \ell_1 \)-norm. In fact the regularization term \( \lambda \sum_{g=1}^G \|\beta_g\|_2 \) is the combination of both the \( \ell_1 \)-norm (inter-group) and the \( \ell_2 \)-norm (intra-group) and thus can be called the \( \ell_{2,1} \)-norm. The fact that the Group Lasso considers multiple elements as a whole implies that it utilizes implicit relations among these elements to some extent. Nevertheless, in order to more precisely characterize the correlations we intend to explicitly integrate spatial correlations among groups of basic regions into the Group Lasso. Another restriction of the Group Lasso is that it can only encode one region at a time, which may lead to
the loss of the intrinsic correlations and consistency among the test regions.

3.1.3. Joint group lasso

Considering a group of test regions segmented from a test image, surely we can reconstruct each region individually to fulfill the local image tagging task for the test image. Although the Group Lasso provides image-level sparsity for individual region reconstruction, it cannot guarantee consistency and robustness in reconstructing all the test regions with the test image. In fact, these reconstructed regions are naturally correlated with each other. Thus, to facilitate a joint region reconstruction, we further propose a joint group lasso model. The joint group lasso helps achieve the following goal: once a group of training regions have been chosen as sparse codes for one test region, then probably should they be chosen to represent other test regions within the same image without adding much extra penalty cost.

Denote \( Y = \{y_1, y_2, \ldots, y_{N_t}\} \in \mathbb{R}^{d \times N_t} \) as \( N_t \) test regions segmented from a test image \( x \). In the previous works [15,2], the authors proposed a simultaneous sparsity regularizer for jointly reconstructing \( Y \) as follows:

\[
\min_B \frac{1}{2} \| Y - XB \|_F^2 + \lambda \sum_{l=1}^{N} \| b_l \|_F
\]

(3)

where \( \| \cdot \|_F \) is defined as the Frobenius norm. The first term penalizes the whole reconstruction error of all encoding regions. \( B = [b_1^T, b_2^T, \ldots, b_N^T] \in \mathbb{R}^{N \times N_t} \) denotes the reconstruction coefficient matrix and \( b_i \in \mathbb{R}^{1 \times N_t} \) (the \( i \)th row of matrix \( B \)) specifies the contribution of the \( i \)th basic region of \( X \) to each test region. In essence, (3) is the joint version of the Lasso. The regularizer \( \lambda \sum_{l=1}^{N} \| b_l \|_F \) tends to minimize the number of nonzero rows in \( B \). As with the Lasso, this formulation ignores correlations among basic elements and only guarantees element-level sparsity rather than group-level sparsity.

To overcome these drawbacks we generalize the model (3) to the group version which guarantees sparsity when jointly reconstructing all the test regions in the same image

\[
\min_B \frac{1}{2} \| Y - XB \|_F^2 + \lambda \sum_{g=1}^{G} \| B_g \|_F
\]

(4)

where \( B = [B_1^T, B_2^T, \ldots, B_N^T] \) and each \( B_g \) is the sparse code submatrix corresponding to the \( g \)th group. As we can see, the joint group lasso in (4) is the natural extension of both (2) and (3). Thus, it helps robustly reconstruct all the test regions at the same time.

3.2. Region reconstruction with local structure preservation

The joint group lasso has been shown to be able to significantly improve the local tagging performance in [41]. But the correlations among the test regions are not explicitly utilized, in other words, the local correlations or structural information are not well preserved. In this part, we bring in a Laplacian prior to handle this problem.

Recall the sparse code matrix \( B = [b_1^T, b_2^T, \ldots, b_N^T] \) and each \( b_g \) corresponds to the sparse codes of the \( g \)th reconstructed region. These sparse codes can also be regarded as the new representations in the sparse coding space. Therefore, to preserve the local structural information and enforce similarity consistency, here we naturally assume that if two regions are similar in the original space, they are supposed to be similar in the new space. This can be encouraged by the following model:

\[
\min_B \sum_{ij=1}^{N} \| b_i^2 - b_j^2 \|_2 W_{ij}
\]

(5)

where \( W \) is the similarity matrix of test regions in the original space and \( W_{ij} \) is defined as follows:

\[
W_{ij} = \begin{cases} 
\exp \left( -\frac{\| y_i - y_j \|_2^2}{2\sigma^2} \right) & \text{if } y_i \in \mathcal{N}_k(y_j) \text{ or } y_j \in \mathcal{N}_k(y_i), \\
0 & \text{otherwise.}
\end{cases}
\]

where \( \sigma \) is the bandwidth parameter and \( \mathcal{N}_k(\cdot) \) is the function of search for the \( k \) nearest neighbors.

By defining a diagonal matrix \( D \) with the \( i \)th diagonal element \( D_{ii} = \sum_{j} W_{ij} \), we can rewrite Eq. (5) as below

\[
\min_B \frac{1}{2} \| Y - XB \|_F^2 + \lambda \sum_{g=1}^{G} \| B_g \|_F + \mu \text{tr}(BLB^T)
\]

(6)

So far we have found a way of linearly reconstructing a group of test regions at the same time with the local structural information preserved. In the next part, we propose to extend the LJGL model to handle the non-linear reconstruction in case the linear assumption does not hold.

3.3. Non-linear region reconstruction

In reality, the basic linear assumption of the sparse coding does not always hold. Kernel methods are usually employed in non-linear scenarios. Therefore, it is natural to kernelize our LJGL model to cope with the non-linear region reconstruction.

Suppose there is a kernel mapping function \( \phi : X \rightarrow \phi(X) \), which is able to project the feature of a test region to a high-dimensional reproducing kernel Hilbert space \( \mathcal{H} \). Thus, we can reconstruct the test region in the kernel space \( \mathcal{H} \) to achieve the non-linear reconstruction

\[
\min_B \frac{1}{2} \| \phi(Y) - \phi(X) B \|_F^2 + \lambda \sum_{g=1}^{G} \| B_g \|_F + \mu \text{tr}(BLB^T)
\]

(7)

where \( \phi(Y) = [\phi(y_1), \phi(y_2), \ldots, \phi(y_{N_t})] \) is the mapped representations for all the test regions and \( \phi(X) = [\phi(x_1), \ldots, \phi(x_k), \ldots, \phi(x_{N_t})] \) is the mapped dictionary for reconstruction. Note that actually the LJGL is a special case of the kernel LJGL if we choose to use Linear Kernel \( \phi : x \rightarrow x \). In the next section, we devise an effective algorithm for globally minimizing the value of the objective function in Eq. (8).

3.4. Optimization

We design an iterative algorithm for optimizing the LJGL model effectively. In each iteration, the update of the sparse code \( y_i \) is defined as the optimal result of the \( t \)th iteration and \( B \) as the variable of \( (t+1) \)th iteration of Algorithm 1, then the following
inequality holds:

\[ \|B\|_F^2 - \frac{\text{tr}(B^T B)}{2\|B\|_F} \leq \|B\|_F^2 - \frac{\text{tr}(\tilde{B}^T \tilde{B})}{2\|B\|_F} \]

Proof. See Appendix.

**Lemma 2.** Given \( B = [b_1^T, b_2^T, \ldots, b_G^T]^T \), \( B_g \) is a sub-matrix of \( B \) and corresponds to the \( g \)th group, then we have the following conclusion:

\[ \sum_{g=1}^{G} \|B_g\|_F^2 - \frac{C}{2\|B\|_F} \leq \sum_{g=1}^{G} \|B_g\|_F^2 - \frac{c_g \text{tr}(B_g^T B_g)}{2\|B\|_F} \]

Proof. See Appendix.

**Algorithm 1.** An effective iterative algorithm for optimizing the LJGL model.

**Input:** A dictionary \( X \) and a test region matrix \( Y \); **Output:** The globally optimal reconstruction coefficients;

1. Construct the kernel matrices \( K_{xx} \) and \( K_{xy} \);
2. Construct the Laplacian matrix \( L \) for the test regions \( Y \);
3. Initialize \( B \);
4. repeat
5. Let \( D = \begin{bmatrix} \|B_1\|_F & \cdots & \|B_G\|_F \end{bmatrix} \);
6. Solve the Sylvester Equation \( (K_{xx} + \lambda D^{-1})B + 2BL = K_{xy} \) to obtain the updated \( B \);
7. until there is no change to \( B \);
8. return \( B \);

With the above two conclusions, we come out with the conclusion of Theorem 3, which guarantees the monotone decreasing of the objective function value.

**Theorem 3.** At each iteration of Algorithm 1, the value of the objective function in Eq. (8) monotonically decreases.

Proof. We first start from optimizing the following quadratic problem w.r.t. \( B \):

\[ \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}\|_F^2 + \lambda \sum_{g=1}^{G} \text{tr}(B_g^T B_g) + \mu \text{tr}(BLB^T) \]

By defining a diagonal matrix \( D \) as below

\[ D = \begin{bmatrix} \|B_1\|_F & \cdots & \|B_G\|_F \end{bmatrix} \]

It is orderly formed by \( G \) sub-diagonal matrices corresponding to \( G \) groups. Thus, we can rewrite Eq. (9) as follows:

\[ \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}\|_F^2 + \lambda \sum_{g=1}^{G} \frac{\text{tr}(B_g^T B_g)}{2\|B_g\|_F} + \mu \text{tr}(BLB^T) \]

\[ \Rightarrow \frac{1}{2} \text{tr}(\phi(Y) - \phi(X)\tilde{B})^T (\phi(Y) - \phi(X)\tilde{B}) + \frac{1}{2} \text{tr}(B^T D^{-1} B) + \mu \text{tr}(BLB^T) \]

\[ \Rightarrow \frac{1}{2} \text{tr}(-2\tilde{B}^T K_{xy} + B^T (K_{xx} + \lambda D^{-1}) B + \mu \text{tr}(BLB^T) \]

where \( K_{xx} = \phi(X)^T \phi(X) \) and \( K_{xy} = \phi(X)^T \phi(Y) \) By setting the deviation of (10) w.r.t. \( \tilde{B} \) to zero we obtain the following equation:

\( (K_{xx} + \lambda D^{-1})\tilde{B}^* + 2\tilde{B}^* L = K_{xy} \)

This is a typical Sylvester Equation [1] which can be solved efficiently by existing optimization toolboxes. Then we substitute both \( \tilde{B}^* \) and \( B \) into (9) and obtain

\[ \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}^*\|_F^2 + \lambda \sum_{g=1}^{G} \frac{\text{tr}(B_g^T B_g)}{2\|B_g\|_F} + \mu \text{tr}(\tilde{B}^* L\tilde{B}^*) \]

\[ \leq \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}\|_F^2 + \lambda \sum_{g=1}^{G} \frac{\text{tr}(B_g^T B_g)}{2\|B_g\|_F} + \mu \text{tr}(BLB^T) \]

\[ \Rightarrow \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}\|_F^2 + \lambda \sum_{g=1}^{G} \frac{\text{tr}(B_g^T B_g)}{2\|B_g\|_F} + \mu \text{tr}(\tilde{B}^* L\tilde{B}^*) \]

\[ \leq \frac{1}{2} \|\phi(Y) - \phi(X)\tilde{B}\|_F^2 + \lambda \sum_{g=1}^{G} \|B_g\|_F + \mu \text{tr}(BLB^T) \]

As we can see, in the \((t+1)\)th iteration the optimal \( \tilde{B}^* \) indeed makes the value of Eq. (8) decreased. \( \square \)

Thanks to the convexity of the objective function in Eq. (8), Theorem 3 clearly guarantees that Algorithm 1 converges to the global optimum.

3.5. Local image tagging

In the previous works [13,41], the reconstruction dictionary for a test image is composed of the regions segmented from all the training images. Nevertheless, this way may suffer from the following shortcomings: on the one hand, the tagging efficiency is a quite challenging issue due to the large scale of the dictionary; on the other hand, tagging noise would be inevitably introduced by those irrelevant image regions. In this section, we present an efficient way of constructing dictionary and integrate it to the local image tagging approach based on the proposed LJGL model.

As aforementioned, it is observed that an image is normally related to a small group of images within an image repository. In other words, most of the images in the repository may not be helpful in the reconstruction process. Therefore, give a test image \( x \), it is reasonable to create its reconstruction dictionary with a small number of potentially relevant images rather than the whole image database. To this end, we first identify the visually nearest neighbors \( N_k(x) \) of the test image using image-level matching. The segmented regions of these neighbors are concatenated to form a dictionary for the test image. The regions segmented from the test image are fed into our proposed LJGL model together with the corresponding dictionary for joint reconstruction. After that, we obtain a group of optimal reconstruction coefficients which indicate the visual importance of the training regions to the test regions. The whole procedure is summarized in Algorithm 2.

**Algorithm 2.** The local image tagging procedure via the LJGL model.

**Input:** An segmented image dataset with region-level tags \( T \) and a test image \( x \);

**Output:** Ranked Tag Lists for segmented regions in \( x \);

1. Find \( k \) nearest neighbors \( N_k(x) \) for the test image \( x \);
2. Segment the test image \( x \) into several regions \( Y \);
3: Construct the dictionary X using the segmented regions of $X_d(y)$;
4: Perform Algorithm 1 to obtain the optimal reconstruction coefficients $T^*$ for all the test regions $Y$;
5: for each test region in $Y$ do
6: Perform Algorithm 3 to propagate tags from the training regions to the test region;
7: end for

Tag propagation has been widely studied in multimedia area, especially for multimedia annotation [17,32]. Inspired by the works in [26,31,30], we propose to propagate tags from the training regions to the test regions based on the following basic assumption, i.e., if a given region can be reconstructed by its visually relevant regions linearly, then its labels can also be reconstructed by the tags of these relevant regions with the same importance. Given a test region $y$, after reconstructing it by the proposed LJGL model, we obtain the sparse coefficient $B = [b_1, b_2, \ldots, b_N]$ corresponding to the training regions, where $b_i$ quantifies the closeness between the $i$th training region and $y$. This means the tag of the $i$th training region can be propagated to $y$ with the weight $b_i$. Denote the tags associated with all the training regions as $T = [t_1, t_2, \ldots, t_N]$, we summarize the tag propagation procedure in Algorithm 3. Note that we only consider the training regions that are able to provide positive contribution in the tag propagation process. As shown, given the test region, we calculate a weight for each unique tag by summing up the coefficients of the training regions associated with this tag. All the unique tags are ranked according to the descending order of their weights. The tag with the highest weight is the most possible tag for the test region.

Algorithm 3. Algorithm for tag propagation.

Input: Sparse code $B = [b_1, b_2, \ldots, b_N]$ and tags $T = [t_1, t_2, \ldots, t_N]$ corresponding to $N$ training regions;
Output: A ranked tag list $C$;
1: Initialize tag list $C = {}$;
2: for $i = 1$ in $N$ do
3: if $b_i > 0$ then
4: if $t_i \in C$ then
5: $C(t_i).weight + = b_i$;
6: else
7: $C.tag = t_i$;
8: $c.weight = b_i$;
9: $C = C \cup \{c\}$;
10: end if
11: end if
12: end for
13: Sort $C$ according to the descending order of $C.weight$;
14: return $C$;

3.6. Algorithm discussion and complexity analysis

Computational cost is of vital importance to the applications of an algorithm, especially the scalability. In this part, we briefly discuss and analyze the complexity of our approach. In each iteration of Algorithm 1, two operations are performed: (1) the construction of $D$, which is a diagonal matrix and can be efficiently solved and (2) updating $B$ by solving the Sylvester Equation

$$(K_{xy} + \epsilon D^{-1})B^* + 2B^*L = K_{xy}$$

which can be achieved via the Bartels–Stewart algorithm [1]. The computational cost of the whole algorithm is apparently dominated by the cost of solving the Sylvester Equation, which can be done with $O(N^3 + N_1^2 + NN^2_2 + N^2_2 N_1)$ operations where $N$ is the number of the training regions in the dictionary and $N_1$ is the number of test regions segmented from a test image. Note that in our application the average number of regions segmented one image is usually limited, which implies that normally $N_1 \ll N$ and $N$ is proportional to $G$, which is the number of training images for constructing the dictionary. Besides the empirical results show that the convergence can be achieved quickly within only a few iterations. Therefore, the computational cost of Algorithm 1 can be roughly summarized as $O(G^3)$. Apparently the algorithm is not suitable to handle large-scale datasets if $G$ is too large. To address this problem, we only employ $k$ nearest neighbors for constructing the dictionary where $k \ll G$, thus the computational cost of our proposed approach is $O(k^3)$, which is affordable if $k$ is carefully chosen.

4. Experiments

In this section, we employ four public image datasets to evaluate our proposed local image tagging method.

4.1. Experimental settings

4.1.1. Datasets

Four image datasets with region-level tagging ground-truth are selected to evaluate the effectiveness of our method. MSRC image datasets [23] provide two versions: MSRC_v1 provides 240 images tagged with 13 keywords while MSRC_v2 is comprised of 591 images associated with 23 keywords. Both of them have been manually segmented and labeled at pixel level. Stephen et al. collected an image dataset from LabelMe, MSRC and PASCAL VOC [7]. It provides 715 images and region-level tagging ground-truth with seven labels. The last one is the SAIAPR TC-12 [3], which consists of around 100,000 regions segmented from 20,000 images. It also provides 275 labels for region-level ground truth. The SAIAPR dataset is organized into 41 subsets and each subset contains relatively relevant images (e.g., images taken at the same landscape). In our experiments, we use the whole SAIAPR and two subsets with 251 images (SAIAPR_SUB1) and 530 images (SAIAPR_SUB2) respectively.

4.1.2. Image segmentation and feature extraction

The proposed method needs to preliminarily segment the images and extract visual features from segmented regions. Various image segmentation algorithms [22,39] can be used here. We use Normalized Cuts Clustering [22] because of its well-known effectiveness in image segmentation. We cannot expect Normalized Cuts to generate the same segmentation results as manual ground truths. Therefore, we assign each region with its dominant label.

For all MSRC, SAIAPR and Stephen’s datasets, we characterize the visual content of their images and regions by extracting Local Binary Patterns (LBP) feature [19]. LBP assigns each pixel with a value by comparing its eight neighbor pixels with the center pixel value and transforming the result to a binary value. Then the histogram of the values is accumulated as a local descriptor.

4.2. Effects of kernels

In this section, we test the effects of different types of kernels on the local image tagging performance.

We choose eight types of kernels, including Linear Kernel, Gaussian Kernel, Laplacian Kernel, Inverse Squared Distance (ISD) Kernel, Inverse Distance (ID) Kernel, Polynomial Kernel, Tangent Kernel and Triangular Kernel. We consistently fix both balance parameters $\lambda$ and $\mu$ in the LJGL model to 0.1 and the number of
nearest neighbors in $k$ NN to 50. For each kernel, we test different kernel parameters and report the best result. For example, in Gaussian Kernel, the bandwidth parameter is set as $\{10^{-6}, 10^{-4}, \ldots, 10^4, 10^6\}$. Apart from the MSRC, Stephen's and the two subsets of SAIAPR mentioned before, we additionally test the kernel effects over the whole SAIAPR. In order to obtain comparable results, we choose to use the images in SAIAPR_SUB1 and SAIAPR_SUB2 for test, and the whole SAIAPR dataset is used to construct the dictionary rather than the subsets themselves. Thus, we obtain two more rows of results (SAIAPR_SUB1_ALL and SAIAPR_SUB2_ALL).

The comparing results are illustrated in Table 1, from which we obtain the following conclusions: (1) Polynomial Kernel and Triangular Kernel always obtain the top two tagging performance among all the eight kernel types. Recall that the kernel LJGL model with Linear Kernel is equivalent to the original LJGL. Thus, this clearly indicates the proposed non-linear kernel extension really helps achieve a better reconstruction. (2) In most cases, distance-based kernels (Gaussian, Laplacian, ISD and ID) perform worse than Linear Kernel. This shows that kernel methods do not necessarily cause positive effects on the local tagging performance. (3) The performances of SAIAPR_SUB1 and SAIAPR_SUB2 are consistently better than those of SAIAPR_SUB1_ALL and SAIAPR_SUB2_ALL respectively. It is possible to find more precise nearest neighbors for a test image in the whole dataset, but it does not lead to a better reconstruction. One possible explanation is that the subsets in SAIAPR are already well organized (e.g., images are taken at the same landscape), and global similarity match does not necessarily guarantee to find the most “relevant” images.

In summary, different types of kernels impose different effects on the local tagging performance. It is necessary to introduce certain proper kernels for better region reconstruction. In the following experiments, we choose to use three kernels with top three results, namely Linear Kernel, Triangular Kernel and Polynomial Kernel.

4.3. Effects of $k$

The purpose of the hierarchical strategy is to improve the efficiency of the region reconstruction as well as avoid noisy tags. In this section, we test the effects of different numbers of nearest neighbors to find a proper $k$ and illustrate the effectiveness of the hierarchical tagging method.

Intuitively, a small number of nearest neighbors would not contain enough tag information, thereby degrading the local tagging performance, while a large $k$ may bring in more potentially noisy tags, which could bring the tagging performance down as well. By default, we set $k$ in the range of $\{25, 50, \ldots, 150\}$. The average accuracies of the effects of $k$ on different datasets are illustrated in Fig. 2. The following observations are found based on these results: (1) In all cases, as $k$ becomes larger most of the curves can see a rising trend. For example, in (b) and (c), when $k$ is from 25 to 100 the average accuracies keep increasing. The clearly shows that a relatively large $k$ might help improve the performance. (2) As $k$ continues to increase, most of the tagging average accuracy curves stop going up. In (a), starting from $k=50$, the Polynomial Kernel dramatically plunges until $k=150$, while Triangular Kernel and Linear Kernel keep relatively stable. In (d), we can see a clear dropping trend in the curve of Polynomial Kernel after $k=100$. In (e) and (f), the performance of Linear Kernel and Triangular Kernel both start to decrease from $k=75$. This gives us an obvious hint that a larger $k$ does not necessarily pose positive effects, but rather introduces more tag noises to degrade the tagging performance. Moreover, a large $k$ definitely degrades the tagging efficiency.

In conclusion, either a too small $k$ or a too large $k$ would not produce the best benefit to the performance of the hierarchical local image tagging. A proper $k$ should be found to achieve a trade-off between the efficiency and the tagging noises. In the following part, we fix $k=50$ for MSRC_v1 and Stephen’s, $k=75$ for SAIAPR_SUB2, SAIAPR_SUB1_ALL and SAIAPR_SUB2_ALL, and $k=100$ for MSRC_v2 and SAIAPR_SUB1.

4.4. Comparison and analysis

We choose to compare with two baseline algorithms (Lasso and Group Lasso) and two state-of-the-art local tagging methods (Bi-Layer sparse coding [13] and joint group lasso [41]). We compare them with the kernel LJGL of three kernels mentioned in the above section. For Lasso, Group Lasso and Bi-Layer Lasso, the sparsity parameter $\lambda$ is set to $[0.1]$ and SLEP package [12] is selected to implement them. For three of the kernel LJGL methods, we have an additional parameter $\mu$ to balance the effect of the Laplacian prior. It is set from $10^{-6}$ to $10^2$. For all comparing methods, the best tagging performances are reported.

The overall comparing results are illustrated in Table 2 and some exemplar results are given in Fig. 3. We have the following observations: (1) In all cases, our proposed kernel LJGL model is able to consistently achieve the best performances. For example, when evaluating on MSRC_v1 and MSRC_v2, LJGL_Poly outperforms Lasso, Group Lasso and Bi-Layer Lasso by around 14% and 10% respectively. This clearly shows the superiority of our propose model. (2) The three LJGL models consistently perform better than the joint group lasso. Clearly, this indicates the effectiveness of the Laplacian prior for similarity preservation. (3) Unlike the results in Section 4.2, LJGL_Poly does not always gain the best results among the three kernel types. In MSRC_v2 and SAIAPR_SUB2_ALL, LJGL_Triangular and LJGL_Linear perform the best respectively. We believe that this is due to the different properties of image datasets. (4) Similar to the results in Section 4.2, the performances of SAIAPR_SUB1 and SAIAPR_SUB2 are consistently better than those of SAIAPR_SUB1_ALL and SAIAPR_SUB2_ALL respectively. This further proves that the organization of subsets in SAIAPR is already good enough to find “relevant” images for the test image. (5) For MSRC_v1, MSRC_v2 and SAIAPR_SUB1, Lasso, Group Lasso and Bi-Layer Lasso always
Fig. 2. Effects of the number of nearest neighbors in the hierarchical local image tagging: (a) MSRC_v1, (b) MSRC_v2, (c) Stephen’s, (d) SAIAPR_SUB1, (e) SAIAPR_SUB2, (f) SAIAPR_SUB1_ALL and (g) SAIAPR_SUB2_ALL.

Table 2
Overall average accuracies of all comparing algorithms.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Lasso</th>
<th>GL</th>
<th>Bi-Layer</th>
<th>JGL</th>
<th>LJGL_Linear</th>
<th>LJGL_Triangular</th>
<th>LJGL_Poly</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC_v1</td>
<td>65.9</td>
<td>65.9</td>
<td>65.9</td>
<td>79.5</td>
<td>79.5</td>
<td>79.3</td>
<td><strong>80.0</strong></td>
</tr>
<tr>
<td>MSRC_v2</td>
<td>69.5</td>
<td>69.4</td>
<td>69.5</td>
<td>79.8</td>
<td>79.8</td>
<td>79.8</td>
<td>79.8</td>
</tr>
<tr>
<td>Stephen’s</td>
<td>74.9</td>
<td>74.9</td>
<td>74.9</td>
<td>80.4</td>
<td>80.6</td>
<td><strong>80.7</strong></td>
<td>80.3</td>
</tr>
<tr>
<td>SAIAPR_SUB1</td>
<td>44.9</td>
<td><strong>50.6</strong></td>
<td>44.9</td>
<td>48.2</td>
<td>49.5</td>
<td>38.6</td>
<td><strong>50.6</strong></td>
</tr>
<tr>
<td>SAIAPR_SUB2</td>
<td>36.0</td>
<td>36.0</td>
<td>36.0</td>
<td>38.5</td>
<td>38.6</td>
<td>38.6</td>
<td><strong>39.6</strong></td>
</tr>
<tr>
<td>SAIAPR_SUB1_ALL</td>
<td>41.7</td>
<td>41.8</td>
<td>41.7</td>
<td>40.0</td>
<td>40.3</td>
<td>40.6</td>
<td><strong>42.5</strong></td>
</tr>
<tr>
<td>SAIAPR_SUB2_ALL</td>
<td>35.5</td>
<td>36.1</td>
<td>35.5</td>
<td>35.9</td>
<td><strong>37.4</strong></td>
<td>37.3</td>
<td>37.0</td>
</tr>
</tbody>
</table>
obtain better results than their counterparts in [41]. We have reasons to believe that the hierarchical strategy really helps reduce the tagging noise.

4.5. Effects of image segmentation

Existing image segmentation methods usually cannot provide satisfactory performance. It is important for us to test the error tolerance of our method to image segmentation. In this part we compare Normalized Cut and Manual Segmentation in terms of the performance for our local image tagging. For Normalized Cut, the number of image segments for each individual image is set to the average number of tags associated with images in the dataset. We use the above two segmentation methods on all the five datasets. To simplify the comparison, we choose the Polynomial Kernel, and consistently fix both balance parameters $\lambda$ and $\mu$ in the LJGL model to 0.1 and the number of nearest neighbors in kNN to 50.

The comparison results are shown in Fig. 4. Generally, the difference between Normalized Cuts and Manual Segmentation is not obvious. For MSRC_v1 and SAIAPR_SUB_2, the Manual Segmentation achieves slightly better performances compared with Normalized Cuts, which implies that Normalized Cuts may not provide as perfect segmentation results as the Manual Segmentation. For MSRC_v2, Stephen’s and SAIAPR_SUB_1, the Normalized Cuts almost achieves the same performances as the Manual Segmentation. In fact, the tagging accuracy is related to segmentation but our model is shown to be robust with respect to the automatic segmentation. Indeed the impact of the imperfect automatic segmentation may degrade the performance of our method under some circumstances, but we believe that even Manual Segmentation is still not perfect enough to support our method to achieve satisfactory performances. Basically, the performance of Normalized Cuts is comparable to that of the Manual Segmentation, which implies that our proposed model has good tolerance to Normalized Cuts.

4.6. Robustness

In practice, the tags of training samples are often noisy, which may severely affect tagging performance. In this part, we investigate the robustness of our approach with the presence of noisy tags. MSRC_v1 and MSRC_v2 are used in this experiment. (5%,10%,...,50%) samples in each dataset are randomly selected and manually mislabeled. Similarly, we choose the Polynomial Kernel, and consistently fix both balance parameters $\lambda$ and $\mu$ to 0.1 and k=50 in kNN search. The results are shown in Fig. 5(a) and (b). As seen, as the noise level varies from 5% to 30%, both Fig. 5(a) and (b) see only slight performance degradation. This gives us a clear hint that our approach is robust to noisy tags.

5. Conclusion and future work

In this paper we extend joint group lasso with a similarity preservation constraint to propose a novel Laplacian Joint Group Lasso (LJGL) model. To achieve efficient and noise-proof region reconstruction, we also design a hierarchical local image tagging method. A set of test regions are able to be jointly reconstructed by the LJGL, and tags are propagated from sparsely selected regions to
the test regions with the reconstruction coefficients. We conduct extensive experiments on four public image datasets to demonstrate the superiority of our proposed model and method.

Our proposal mainly relies on an existing training dataset, which makes it an interesting issue as to how to handle emerging images on the Web. In the future we will focus on improving the ability of coping with such new Web images. For instance, we may construct the dictionary by gathering training images from general image search engines.

Appendix A. Proof of Lemmas 1 and 2

First we prove Lemma 1.

Proof. Since $\|B_{g}\|_{F}$ and $\|B_{f}\|_{F}$ are real values, we have

\[\begin{align*}
\langle \|B_{g}\|_{F} - \|B_{f}\|_{F} \rangle^2 &\geq 0 \\
\Rightarrow \text{tr}(B_{g}^T B_{g}) + \text{tr}(B_{f}^T B_{f}) - 2\|B_{g}\|_{F} \|B_{f}\|_{F} &\geq 0 \\
\Rightarrow 2\|B_{g}\|_{F} - \text{tr}(B_{g}^T B_{g}) &\leq \frac{\text{tr}(B_{f}^T B_{f})}{\|B_{f}\|_{F}} \\
\Rightarrow \|B_{g}\|_{F} &\leq \frac{\text{tr}(B_{f}^T B_{f})}{\|B_{f}\|_{F}} - \frac{\text{tr}(B_{g}^T B_{g})}{\|B_{f}\|_{F}} \tag{1}
\end{align*}\]

Next we prove Lemma 2.

Proof. According to Lemma 1, for each group $g$ we have

\[\|B_{g}\|_{F} \leq \frac{\text{tr}(B_{g}^T B_{g})}{\|B_{f}\|_{F}} \leq \frac{1}{\|B_{f}\|_{F}} \|B_{f}\|_{F} - \frac{\text{tr}(B_{g}^T B_{g})}{\|B_{f}\|_{F}} \|B_{f}\|_{F} \]

Thus, by summing over all $G$ above inequalities we get the conclusion of Lemma 2. \hfill \Box

Appendix B. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2012.10.026.

References


Yang Yang is a PhD candidate in School of Information Technology & Electrical Engineering, The University of Queensland. His research topics include content understanding, pattern recognition, multimedia tagging/annotation and multimedia retrieval.

Zi Huang is a Lecturer and an Australian Postdoctoral Fellow in School of Information Technology & Electrical Engineering, The University of Queensland. She received her BSc degree from Department of Computer Science, Tsinghua University, China, and her PhD in Computer Science from School of ITEE, The University of Queensland. Dr. Huang’s research interests include multimedia search and knowledge discovery.

Yi Yang is a postdoctoral research fellow in School of Computer Science, Carnegie Mellon University. He worked for the University of Queensland as a postdoctoral research fellow from Sep 2010 to May 2011. After that, he joined the School of Computer Science at Carnegie Mellon University. His research interests include machine learning and its applications to multimedia content analysis and computer vision, e.g., multimedia indexing and retrieval, image annotation, video semantics understanding, etc.

Jiajun Liu is a PhD candidate in School of Information Technology & Electrical Engineering, The University of Queensland. His research topics include multimedia search and multi-core concurrent programming.

Heng Tao Shen is a Professor of Computer Science in School of Information Technology & Electrical Engineering, The University of Queensland. He obtained his BSc (with 1st class Honours) and PhD from Department of Computer Science, National University of Singapore in 2000 and 2004 respectively. His research interests include multimedia/mobile/web search, database management, dimensionality reduction, etc. Heng Tao has extensively published and served on program committees in most prestigious international publication venues in multimedia and database societies. He is also the winner of Chris Wallace Award for outstanding Research Contribution in 2010 from CORE Australasia. He is a Senior Member of IEEE and Member of ACM.

Jiebo Luo is a Professor in Department of Computer Science in the University of Rochester. His research spans image processing, computer vision, machine learning, data mining, medical imaging and ubiquitous computing. He has been an Advocate for contextual inference in semantic understanding of visual data, and continues to push the frontiers in this area by incorporating geo-location context and social context. He has published extensively with over 180 papers and 60 US patents. He has been involved in numerous technical conferences, including serving as the program co-chair of ACM Multimedia 2010 and IEEE CVPR 2012. He is the Editor-in-Chief of the Journal of Multimedia, and has served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, Pattern Recognition, Machine Vision and Applications and Journal of Electronic Imaging. He is a Fellow of the SPIE, IEEE and IAPR.