Computer Vision and Image Understanding 124 (2014) 42–60

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Image tag completion via dual-view linear sparse reconstructions

Zijia Lin [a,*], Guiguang Ding [b], Mingqing Hu [c], Yunzhen Lin [b], Shuzhi Sam Ge [d]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
[b] School of Software, Tsinghua University, Beijing 100084, PR China
[c] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, PR China
[d] Department of Electrical and Computer Engineering, The National University of Singapore, Singapore 117576, Singapore

ARTICLE INFO

Article history:
Received 15 August 2013
Accepted 27 March 2014

Keywords:
Image tag completion
Linear sparse reconstruction
DLSR
Image tagging
Tag refinement

ABSTRACT

User-provided textual tags of web images are widely utilized for facilitating image management and retrieval. Yet they are usually incomplete and insufficient to describe the whole semantic content of the corresponding images, resulting in performance degradations of various tag-dependent applications. In this paper, we propose a novel method denoted as DLSR for automatic image tag completion via **D**ual-view **L**inear **S**parse **R**econstructions. Given an incomplete initial tagging matrix with each row representing an image and each column representing a tag, DLSR performs tag completion from both views of image and tag, exploiting various available contextual information. Specifically, for a to-be-completed image, DLSR exploits image-image correlations by linearly reconstructing its low-level image features and initial tagging vector with those of others, and then utilizes them to obtain an image-view reconstructed tagging vector. Meanwhile, by linearly reconstructing the tagging column vector of each tag with those of others, DLSR exploits tag-tag correlations to get a tag-view reconstructed tagging vector with the initially labeled tags. Then both image-view and tag-view reconstructed tagging vectors are combined for better predicting missing related tags. Extensive experiments conducted on benchmark datasets and real-world web images well demonstrate the reasonableness and effectiveness of the proposed DLSR. And it can be utilized to enhance a variety of tag-dependent applications such as image auto-annotation.

1. Introduction

Recently with the prevalence of social network and digital photography, numberless images have been posted to various photo sharing communities, *e.g.* Flickr. Apart from the shared visual information, such large-scale and rapidly-increasing social images are usually associated with user-provided textual tags for describing their corresponding semantic content, which are widely utilized for facilitating kinds of tag-based image applications like text-based image retrieval, *etc*. However, as the manual labeling process can be time-consuming and arbitrary, the user-provided tags probably contain imprecise ones and are usually incomplete, as also revealed in [1,2]. Fig. 1 gives an illustration of the user-provided tags with an exemplary image downloaded from Flickr. From the illustration we can see that the user-provided tags may not only contain misspelling or imprecise ones (*e.g.* “mtn”), but also miss other semantically related ones (*e.g.* “sea”, “water”, “sky” and “grass”).

The imprecision and incompleteness of user-provided tags can lead to performance degradations of various tag-dependent applications. Taking tag-based image retrieval as an example, imprecision of tags will lower the retrieval precision while incompleteness will lower the recall. Therefore, in recent years, tag refinement, including tag denoising and completion, has become an attractive subject of many ongoing researches and has been attracting much attention from both academia and industry. However, previous work on tag refinement, as referred to in related work with details, focused more on denoising but less on completion. As our experiments will show, incompleteness of image tags can bring serious negative effects to tag-dependent applications. And thus we propose that tag completion still deserves further attention and researches, and more effective tag completion methods are expected to be developed.

Given an incomplete initial tagging matrix, with each row representing an image and each column representing a tag, tag completion is to fill it up by identifying more correct associations between images and tags. Specifically, each entry of the initial tagging matrix is either 1 or 0, with 1 indicating that the

[*] Corresponding author.
E-mail addresses: linzijia07@tsinghua.org.cn (Z. Lin), dinggg@tsinghua.edu.cn (G. Ding), humingqing@ict.ac.cn (M. Hu), lin-yz12@mails.tsinghua.edu.cn (Y. Lin), samge@nus.edu.sg (S. Sam Ge).

http://dx.doi.org/10.1016/j.cviu.2014.03.012
1077-3142/© 2014 Elsevier Inc. All rights reserved.

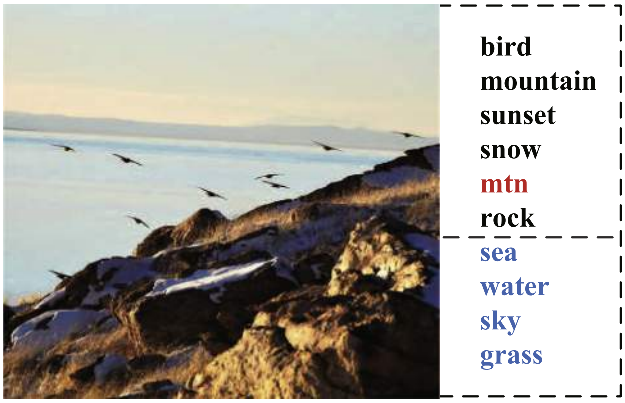


Fig. 1. An exemplary image downloaded from Flickr, with its initially labeled tags (black & red) and several missing related ones (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

corresponding image contains the corresponding tag and 0 otherwise. Then tag completion is essentially to correct false 0 entries into 1 entries.

In this paper, we propose a novel method denoted as DLSR to perform automatic image tag completion via **D**ual-view **L**inear **S**parse **R**econstructions. Specifically, given the initial tagging matrix, the proposed DLSR completes it from both views of image and tag, exploiting various available contextual information. For any to-be-completed image, DLSR exploits image-image correlations by linearly reconstructing its low-level image features and initial tagging vector with those of others, under constraints of sparsity. Then the obtained reconstruction weights are utilized for obtaining an image-view reconstructed tagging vector. Meanwhile, by linearly reconstructing the tagging column vector of each tag with those of others, DLSR exploits tag-tag correlations to get a tag-view reconstructed tagging vector with the initially labeled tags. Then both image-view and tag-view reconstructed tagging vectors are normalized and combined with effective strategies in the field of meta-search to predict the relevance of unlabeled tags to the to-be-completed image. And those with higher relevance are then selected and added.

Instead of performing global refinement for the initial tagging matrix, DLSR performs tag completion via reconstructing each image (*i.e.* row) and each tag (*i.e.* column) separately. And thus it can be utilized to perform tag completion for an unseen image (*i.e.* inductive method) or an existing dataset (*i.e.* transductive method). Specifically, for an unseen image, DLSR only exploits the completely or partially labeled images in the training set to perform tag completion, and thus is used as an inductive method. And for an existing dataset, DLSR performs tag completion for each to-be-completed image in it with all other images, including other to-be-completed images and the training images, since all the to-be-completed images are already observed and also partially labeled, which can probably provide extra helpful information. In this case DLSR is used as a transductive method. DLSR is evaluated with extensive experiments conducted on benchmark datasets and real-world web images. Experimental results well demonstrate its reasonableness and effectiveness. And it can be utilized for enhancing a variety of tag-dependent applications like image auto-annotation, *etc.*

The main contributions of our work can be summarized as follows.

- We propose a novel effective tag completion method via dual-view linear sparse reconstructions, considering and exploiting various available contextual information.
- We propose to perform tag completion via reconstructing each image and each tag separately, instead of performing global refinement for the initial tagging matrix, which enables DLSR to be used as either an inductive method or a transductive one.

This paper is an extension and improvement of our previous work presented in [3]. And we enhance it to be more effective and practical. Specifically, in image-view reconstruction, here we propose to utilize the same reconstruction weights for concurrently reconstructing the low-level features and the initial tagging vector of a to-be-completed image with those of others, in order to simplify model tuning with less parameters while keeping similar performance. Moreover, to prevent the reconstruction weights from being dominating in only images containing an identical initial tagging vector to that of the to-be-completed image, which will provide no information about the missing tags and thus make the image-view reconstruction not work for tag completion, we introduce a “diversity regularizer” in the objective function, as will be elaborated later. Furthermore, to better combine the image-view and the tag-view reconstructed tagging vectors, we propose to treat both as the tag retrieval results from two distinct “search engines”, and resort to effective normalization and combination strategies in the field of meta-search for performance improvement. Experimental results demonstrate that the introduced model enhancements here can generally help to gain performance improvement for the proposed method.

The remainder of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 elaborates on the proposed DLSR, presenting formula details. Then detailed description of experiments, including experimental settings, results and analyses, is given in Section 4. And in Section 5 we investigate various applications that DLSR can be used for. Finally we conclude the paper in Section 6.

2. Related work

As tag completion is to add tags with higher relevance to a given image, it would be natural to compare it with image auto-annotation and tag recommendation. Image auto-annotation [4–10] is to automatically and objectively associate unlabeled images with semantically related tags. Feng et al. [4] proposed a generative learning approach for auto-annotation based on multiple Bernoulli relevance model. Liu et al. [6] built multiple graphical models with various correlations between images and tags, and then performed auto-annotation with manifold learning processes. Makadia et al. [5] proposed a widely-used auto-annotation baseline termed JEC, which is a straightforward greedy algorithm propagating labels from nearest visual neighbors to a to-be-annotated image. And Guillaumin et al. [7] proposed to adopt discriminative metric learning methods in nearest neighbor models, putting forward a state-of-the-art auto-annotation model termed TagProp. In [8,9], Ma et al. further proposed effective methods to exploit the original feature space, via sparsity-based feature selection or uncovering shared subspace, to improve the performance of image auto-annotation. Tag recommendation [2,11–15] is a trade-off between auto-annotation and manual tagging, which is to recommend semantically related tags to a user while he is annotating an image online. Sigurbjörnsson and Zwol [2] proposed a generic tag recommendation method exploiting the collective knowledge residing in images. Wu et al. [12] proposed a learning-based multi-modality recommendation algorithm by considering both tag and visual correlations. And Lee et al. [13] formulated tag recommendation as a maximum a posteriori (MAP) problem using a visual folksonomy.

When comparing tag completion and image auto-annotation, the former can be seen as a special case of the latter. However, many existing auto-annotation methods assume that images in the training set are *completely* labeled with *precise* tags, as also revealed by Qi et al. [16], and they generally focus on predicting tags for fully unlabeled images. Yet for tag completion, images in both training and test sets can all be partially labeled, and thus applying auto-annotation methods to tag completion may not work well. Because their performance can be negatively affected by the partially labeled training set, and they generally neglect to exploit the initial tags of to-be-completed images, which actually can provide important clues for discovering the missing related ones. As for tag recommendation methods, they are generally designed to work online and prefer to interacting with labellers and incorporating feedbacks, while tag completion can be automatically done offline with much looser requirements of real-time performance.

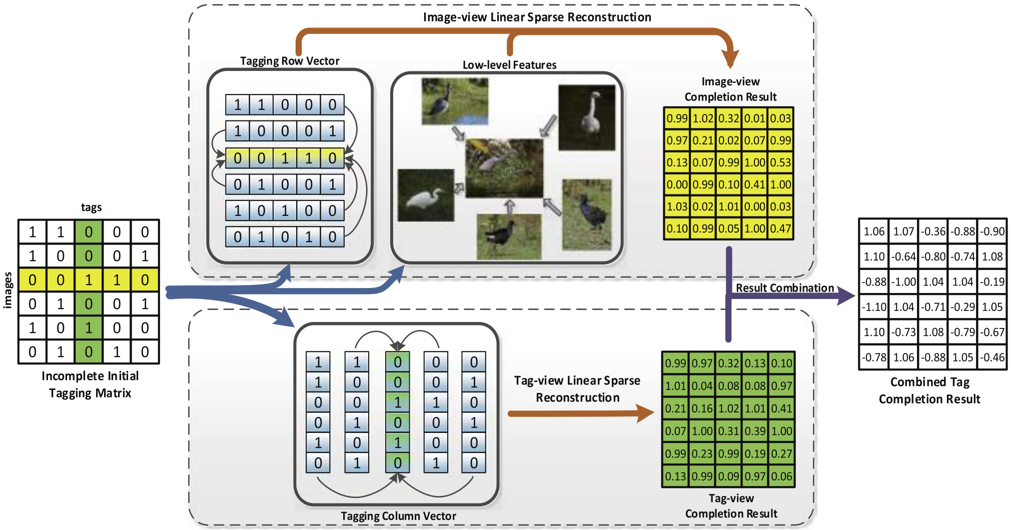
Tag completion is also closely related to tag refinement, which focuses on improving the quality of user-provided tags. Tag refinement includes tag denoising and completion, and has recently become an attractive subject of many ongoing researches [17–25]. As a pioneer work, Jin et al. [17] combined multiple semantic similarity measurements based on WordNet [26] to estimate the correlations between tags, and then removed the weakly-related ones. Xu et al. [19] proposed to distinguish unrelated tags with topic model and further presented regularized Latent Dirichlet Allocation (*i.e.* rLDA) for tag refinement. Lee et al. [21] utilized neighbor voting to learn the relevance of each tag to an image, and then differentiated noisy tags from correct ones. Liu et al. [22] performed tag denoising based on the consistency between "visual similarity" and "semantic similarity" in images, and then enriched the refined tags with their synonyms and hypernyms in WordNet. Zhu et al. [23] formulated the tag refinement problem as a decomposition of the initial tagging matrix into a low-rank refined tagging matrix and a sparse error matrix, with an optimization objective of low-rank, content consistency, tag correlation and error sparsity. Liu et al. [24] proposed to treat each pair of associated tag and image as a semantic unity, and further built a hyper-graph model with semantic unities for tag clustering and refinement.

By reviewing previous researches on tag refinement, we realize that they focused more on tag denoising but less on tag completion. Although a few of them (*i.e.* [23–25]) were claimed to be unified frameworks for both denoising and completion, their performance still needs further improvement. Since the incompleteness of tags can also result in significant performance degradations of tag-dependent applications, as will be validated by our experiments later, we propose that tag completion still deserves further attention and researches, and more effective tag completion methods are expected to be developed. Recently, Wu et al. [27] proposed to address the tag completion problem by searching for the optimal tagging matrix which is consistent with both observed tags and visual similarities. Liu et al. [28] proposed to utilize non-negative data factorization method to perform tag completion, embedding various available contextual information like within-image and cross-image relations, *etc.*

3. Proposed DLSR

An illustration of the framework of the proposed DLSR is given in Fig. 2. It can be seen that DLSR consists of two parts, *i.e.* image-view (upper dotted rectangle) and tag-view (lower dotted rectangle) linear sparse reconstructions. And for each to-be-completed image, they will be separately performed to obtain an image-view reconstructed tagging vector and a tag-view one. Then both will be normalized and combined for predicting the relevance of unlabeled candidate tags. And those with higher relevance will be selected for tag completion.

Specifically, in image-view linear sparse reconstruction, the low-level image features and the initial tagging vector of an image is concurrently reconstructed with those of others. And in tag-view linear sparse reconstruction, the corresponding tagging column vector of a tag in the initial tagging matrix is reconstructed with those of others. Both the image-view and the tag-view reconstructions are formulated as convex optimization problems under constraints of sparsity. The sparsity constraints are attributed to the observation that generally an image contains only a few objects and a tag connotes only a few levels of meaning, and usually the corresponding objects or levels of meaning are redundantly contained or implied in the context. It should be noticed that the tagging vectors utilized in the image-view and the tag-view linear sparse reconstructions are quite different. The former are to exploit the image-image semantic similarities while the latter are to



Incomplete Initial Tagging Matrix

1	1	0	0	0
1	0	0	0	1
0	0	1	1	0
0	1	0	0	1
1	0	1	0	0
0	1	0	1	0

Tagging Row Vector

1	1	0	0	0
1	0	0	0	1
0	0	1	1	0
0	1	0	0	1
1	0	1	0	0
0	1	0	1	0

Tagging Column Vector

1	1	0	0	0
1	0	0	0	1
0	0	1	1	0
0	1	0	0	1
1	0	1	0	0
0	1	0	1	0

Image-view Completion Result

0.99	1.02	0.32	0.01	0.03
0.97	0.21	0.02	0.07	0.99
0.13	0.07	0.99	1.00	0.53
0.00	0.99	0.10	0.41	1.00
1.03	0.02	1.01	0.00	0.03
0.10	0.99	0.05	1.00	0.47

Tag-view Completion Result

0.99	0.97	0.32	0.13	0.10
1.01	0.04	0.08	0.08	0.97
0.21	0.16	1.02	1.01	0.41
0.07	1.00	0.31	0.39	1.00
0.99	0.23	0.99	0.19	0.27
0.13	0.99	0.09	0.97	0.06

Combined Tag Completion Result

1.06	1.07	-0.36	-0.88	-0.90
1.10	-0.64	-0.80	-0.74	1.08
-0.88	-1.00	1.04	1.04	-0.19
-1.10	1.04	-0.71	-0.29	1.05
1.10	-0.73	1.08	-0.79	-0.67
-0.78	1.06	-0.88	1.05	-0.46

Fig. 2. Framework of the proposed DLSR, illustrated with toy data. Given an incomplete initial tagging matrix, DLSR separately performs tag completion via image-view (upper dotted rectangle) and tag-view (lower dotted rectangle) linear sparse reconstructions, and combines the corresponding results for better predicting missing related tags.

exploit the tag-tag correlations. Therefore, the image-view linear sparse reconstruction mainly utilizes the visual similarities and semantic similarities between images, while the tag-view linear sparse reconstruction exploits the correlations between tags. And thus the proposed DLSR considers various available contextual information for tag completion.

With the results of both image-view and tag-view linear sparse reconstructions, the proposed DLSR further combines them for better predicting the missing related tags. In this paper, we propose to treat the image-view and the tag-view results as the tag retrieval results from two distinct "search engines", and resort to effective normalization and combination strategies in the field of meta-search for better combining them and achieving further performance improvement.

3.1. Image-view linear sparse reconstruction

Given a partially labeled dataset, image-view linear sparse reconstruction is to exploit the image-image correlations for obtaining an image-view reconstructed tagging vector for each to-be-completed image. As mentioned previously, both low-level image features and high-level initial tagging vectors are considered. Specifically, for any to-be-completed image I, its feature vector and initial tagging vector are optimally reconstructed with those of others. Following is the framework of the corresponding objective function.

$$\Theta = \min_{\alpha} \; \Theta_1(\alpha) + \mu\Theta_2(\alpha) + \varphi(\alpha) \tag{1}$$

where $\alpha_{k\times1}$ is the required weighting vector consisting of the reconstruction weights of other k images in the image-view linear sparse reconstruction, $\Theta_1(\alpha)$ and $\Theta_2(\alpha)$ are respectively the reconstruction residuals *w.r.t* the feature vector and the initial tagging vector of I, μ is a weighting parameter for balancing $\Theta_1(\alpha)$ and $\Theta_2(\alpha)$, and $\varphi(\alpha)$ is a set of regularizers for α.

Linear sparse reconstruction *w.r.t* low-level image features is to reconstruct an image with others using their corresponding feature vectors. Assuming the feature vector of the to-be-completed image is $\mathbf{f}_{l\times1}$, where l is the dimensionality, the reconstruction residual *w.r.t* low-level image features can be formulated as follows.

$$\Theta_1(\alpha) = \|\mathbf{f} - \mathbf{F}\alpha\|_2^2 \tag{2}$$

where $\|\cdot\|_2$ denotes $l2$ norm, and $\mathbf{F}_{l\times k}$ is a dictionary matrix consisting of feature vectors of the other k images.

Similarly, the linear sparse reconstruction *w.r.t* initial tagging vectors is to reconstruct an image with others using their corresponding initial tagging vectors. The reconstruction residual is formulated in a similar way as follows.

$$\Theta_2(\alpha) = \|\mathbf{W}\left(\mathbf{t} - \widehat{\mathbf{T}}\alpha\right)\|_2^2 \tag{3}$$

where $\mathbf{t}_{n\times1}$ is the n-dimensional initial tagging vector of the to-be-completed image, with n being the number of tags, and $\widehat{\mathbf{T}}_{n\times k}$ is the dictionary matrix consisting of tagging vectors of the other k images. Here $\mathbf{W}_{n\times n}$ is a diagonal matrix for weighting the reconstruction residual of each entry in $\mathbf{t}$, defined as $\mathbf{W}_{i,i} = \exp(\mathbf{t}_i)$. It can be seen that $\mathbf{W}$ assigns higher weights to the non-zero entries (*i.e.* initially labeled tags) of the initial tagging vector $\mathbf{t}$, since they are already ensured while the zero ones (*i.e.* unlabeled tags) are not.

Furthermore, we introduce a sparse group lasso regularizer and a diversity regularizer to the objective function of image-view linear sparse reconstruction. The former regularizer, *i.e.* sparse group lasso, is inspired by the following observations: (1) generally an image contains only a few objects that are redundantly contained in other images, (2) an image is usually associated with only a few tags and images containing an identical tag probably share more common semantic content. The first observation implies that the reconstruction weighting vector α is expected to be sparse. And the second suggests that the non-zero entries of α should correspond to images sharing only a few common tags. Therefore, we propose the following sparse group lasso regularizer $\varphi_1(\alpha)$.

$$\varphi_1(\alpha) = \|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2 \tag{4}$$

Here we introduce a group structure for α as [29], and g_i is the ith group of reconstruction weights, *i.e.* $g_i = \left[\alpha_{\kappa(i,1)}, \alpha_{\kappa(i,2)}, \ldots, \alpha_{\kappa(i,|g_i|)}\right]^T$, where $\kappa(i,j)$ is the index of the jth weight of the ith group in α. Specifically, images sharing a common tag will form a group, and thus each candidate tag corresponds to a group of reconstruction weights, *i.e.* the weights of images containing the tag. Then in formula (4), n is the number of candidate tags. Note that the groups can be overlapped since images are usually labeled with several tags. In $\varphi_1(\alpha)$, the group lasso part, *i.e.* $\sum_{i=1}^{n}\|g_i\|_2$, separately utilizes $l2$ norm for smoothing intra-group weights and $l1$ norm for emphasizing inter-group sparsity. And the lasso part, *i.e.* $\|\alpha\|_1$, further enforces α to be sparse. Therefore, the combined sparse group lasso regularizer can enforce the non-zero entries of α to correspond to only a few images and a few tags, which is expected by the former two observations.

However, the sparse group lasso regularizer may lead the reconstruction weights to being dominating in only a few images containing an identical initial tagging vector to that of the to-be-completed image I, since they are probably more similar to I in both image features and initial tagging vectors. Fig. 3 gives an illustration of the case, where the to-be-completed image I is initially labeled with "clouds", and another two missing related tags, *i.e.* "sun" and "sky", are expected to be completed. With only the sparse group lasso regularizer, the low-level feature vector and the initial tagging vector of I, can be well reconstructed with only image 1 and 2, as they have the same initial tagging vectors and are quite visually similar. However, in that case, the image-view reconstruction cannot provide any helpful information about the missing related tags. Because the image-view reconstructed tagging vector, as will be elaborated in formula (7), is obtained by linearly combining the tagging vectors of images with non-zero reconstruction weights, and thus entries corresponding to the unlabeled tags, including the missing related ones, will be zero. Then in that case, image-view linear sparse reconstruction will not work for tag completion. To avoid that, we introduce a diversity regularizer $\varphi_2(\alpha)$ as follows.

$$\varphi_2(\alpha) = \|\mathbf{s}^T\alpha\|_2^2 \tag{5}$$

where $\mathbf{s}_{k\times1}$ is the non-negative similarities between the initial tagging vector of I and those of the other k images. In our experiments, $\mathbf{s}$ is calculated as the cosine similarities between initial tagging vectors, with all its entries lying in $[0,1]$. Since entries of the objective weighting vector α will mostly be non-negative, we can see that the regularizer will help to penalize the large reconstruction weights of images associated with the same initial tagging vectors as that of I. And thus it can give other visually similar images containing a similar but not identical initial tagging vector, *e.g.* image 3, 4 and 5 in Fig. 3, more chances to contribute to the image-view reconstruction with non-zero weights and then provide more information about the missing related tags.

Therefore, for image-view linear sparse reconstruction considering both low-level image features and high-level initial tagging vectors, we can obtain the integral objective function as follows.

$$\begin{aligned}\Theta = \min_{\alpha}&\|\mathbf{f} - \mathbf{F}\alpha\|_2^2 + \mu\|\mathbf{W}\left(\mathbf{t} - \widehat{\mathbf{T}}\alpha\right)\|_2^2 + \omega\|\mathbf{s}^T\alpha\|_2^2 \\ &+ \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right)\end{aligned} \tag{6}$$

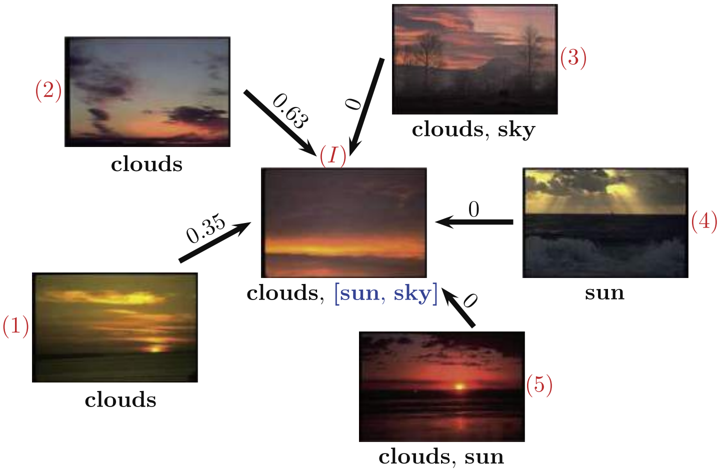


Fig. 3. An illustration on toy data of the case where the sparse group lasso regularizer leads the reconstruction weights to being dominating in only a few images (*i.e.* image 1 and 2) containing the same initial tagging vector as the to-be-completed image *I*. Here tags in black color are the initial tags with images, while the blue ones are the missing related tags of *I*, and values on the arrows are the corresponding reconstruction weights of images in the image-view linear sparse reconstruction for *I*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where λ and ω are weighting parameters for balancing the effects of regularizers. The image-view objective function Θ can be demonstrated to be convex, meaning that there exists a global optimal solution. For details regarding the proof, one can refer to A.1. Then the optimal α can be utilized for obtaining an image-view reconstructed tagging vector $\mathbf{t}_1$ for the to-be-completed image *I*, as shown in formula (7).

$$\mathbf{t}_1 = \widehat{\mathbf{T}}\alpha \tag{7}$$

3.2. Tag-view linear sparse reconstruction

Given a partially labeled dataset, tag-view linear sparse reconstruction is to exploit the tag-tag correlations for obtaining a tag-view reconstructed tagging vector for each to-be-completed image with its initially labeled tags. Specifically, for each tag, its corresponding tagging column vector $\mathbf{r}_{m\times 1}$ in the initial tagging matrix, is linearly reconstructed with those of others as the following formula, with m being the number of images in the given dataset.

$$\Psi = \min_{\beta} \quad \|\mathbf{W}'\left(\mathbf{r} - \widehat{\mathbf{R}}\beta\right)\|_2^2 + \xi\|\beta\|_1 \tag{8}$$

where $\beta_{(n-1)\times 1}$ is the required weighting vector consisting of the reconstruction weights of other tags, $\widehat{\mathbf{R}}_{m\times(n-1)}$ is the dictionary matrix consisting of the tagging column vectors of other tags, and ξ is a weighting parameter for penalizing the non-sparsity of β. Additionally, $\mathbf{W}'_{m\times m}$ is a diagonal weighting matrix for the reconstruction residuals of all entries of $\mathbf{r}$, which is defined in the same way as $\mathbf{W}$ in formula (3).

The tag-view objective function Ψ can be demonstrated to be convex and thus there exists a global optimal solution. For details concerning the proof, one can refer to A.2. Actually, for the hth tag in the vocabulary, by adding a zero entry at its corresponding position in the optimal β, we can obtain $\hat{\beta}_{n\times 1} = [\beta_1, \ldots, \beta_{h-1}, 0, \beta_h, \ldots, \beta_{n-1}]^T$. Then all the obtained $\hat{\beta}$ of each tag can form a tag-tag correlation matrix $\mathcal{B}_{n\times n}$ column by column. And for any given initial tagging vector $\mathbf{t}$ of a to-be-completed image, $\mathcal{B}$ can be utilized to obtain a tag-view reconstructed tagging vector $\mathbf{t}_2$ as follows.

$$\mathbf{t}_2 = \mathcal{B}^T\mathbf{t} \tag{9}$$

Note that when the to-be-completed image is fully unlabeled, *i.e.* entries of $\mathbf{t}$ are all zeros, the tag-view reconstruction will not work, as the obtained $\mathbf{t}_2$ will be a zero-entry vector. Actually in that case, even the image-view reconstruction can only reconstruct the low-level feature vector of the to-be-completed image with those of others, which will probably not work well. Therefore, we suggest that the proposed DLSR is better to be applied to performing tag completion for partially labeled images with at least one initial tag, rather than fully unlabeled ones.

3.3. Combination of image-view and tag-view results

For any to-be-completed image, an image-view reconstructed tagging vector $\mathbf{t}_1$ and a tag-view reconstructed tagging vector $\mathbf{t}_2$ can be respectively obtained with the proposed image-view and tag-view linear sparse reconstructions. Then it would be important to combine both in an appropriate way for better predicting the missing related tags.

In this paper, we propose to treat $\mathbf{t}_1$ and $\mathbf{t}_2$ as the results of retrieving related tags with the given to-be-completed image and its initially labeled tags as a query from two distinct "search engines" (*i.e.* both views of image and tag). Then $\mathbf{t}_1$ and $\mathbf{t}_2$ can be normalized and combined with available effective normalization and combination strategies in the field of meta-search [30–34]. Meta-search is an extensively researched topic focusing on how to combine the retrieval results from different search engines to yield the optimal retrieval performance. As revealed by Vogt and Cottrell [30], a linear combination strategy is generally more flexible than most of others. And thus in our experiments, we utilize the ZMUV (Zero-Mean, Unit-Variance) normalization method proposed by Montague and Aslam [32] to separately normalize $\mathbf{t}_1$ and $\mathbf{t}_2$, and then linearly combine both to be an integral reconstructed tagging vector $\mathbf{t}'$, as shown in the following formula.

$$\mathbf{t}' = \delta\frac{\mathbf{t}_1 - \mu_{\mathbf{t}_1}}{\sigma_{\mathbf{t}_1}} + (1-\delta)\frac{\mathbf{t}_2 - \mu_{\mathbf{t}_2}}{\sigma_{\mathbf{t}_2}} \tag{10}$$

where $\mu_{\mathbf{t}_i}$ and $\sigma_{\mathbf{t}_i}$ are respective the mean value and the standard deviation of $\mathbf{t}_i$ $(i = 1, 2)$, and δ is a weighting parameter in $(0, 1)$ for balancing the results of both views. Then based on $\mathbf{t}'$, which reflects the relevance of tags to the to-be-completed image, unlabeled tags with higher relevance are selected and added.

3.4. Solution and implementation issues

The objective functions of the image-view and the tag-view linear sparse reconstructions, *i.e.* formula (6) and (8), are both unconstrained convex optimization problems. Hence kinds of effective (sub-) gradient descent based methods can be utilized for optimization.

For the objective function of image-view linear sparse reconstruction (*i.e.* formula (6)), as all model parameters (*i.e.* μ, ω, λ) are non-negative, it can be rewritten as follows to be a standard form of *Least Squares Loss Problem* [35] regularized by an overlapping sparse group lasso.

$$\Theta = \min_{\alpha} \|\mathcal{X}_1 \alpha - \mathbf{y}_1\|_2^2 + \lambda \left(\|\alpha\|_1 + \sum_{i=1}^{n} \|g_i\|_2 \right) \tag{11}$$

where

$$\mathcal{X}_1 = \begin{bmatrix} \mathbf{F} \\ \mathbf{W}\widehat{\mathbf{T}}\sqrt{\mu} \\ \mathbf{s}^T\sqrt{\omega} \end{bmatrix}, \qquad \mathbf{y}_1 = \begin{bmatrix} \mathbf{f} \\ \mathbf{W}\mathbf{t}\sqrt{\mu} \\ 0 \end{bmatrix}$$

Then the gradient *w.r.t* α can be derived as follows.

$$\frac{\partial \Theta}{\partial \alpha} = 2\mathcal{X}_1^T \mathcal{X}_1 \alpha - 2\mathcal{X}_1^T \mathbf{y}_1 + \lambda \left(I(\alpha) + \sum_{i=1}^{n} \frac{E^{(i)}\alpha}{\|g_i\|_2} \right) \tag{12}$$

where $I(\alpha)$ is an indicator function for all entries of α, defined as $I(\alpha)_i = \frac{a_i}{|a_i|}$ and assigned as some particular value when $| a_i |= 0$ since $\|\alpha\|_1$ is not differentiable at zero entries [36]. And $E^{(i)}$ is a diagonal indicator matrix with $E^{(i)}_{j,j}$ being 1 if $\alpha_j \in g_i$ and 0 otherwise. For details of the derivation, one can refer to A.1. In our experiments, we utilize the widely-used sparse learning package SLEP [35], to optimize Θ and obtain the optimal α for each to-be-completed image.

Similarly, the objective function of tag-view linear sparse reconstruction (*i.e.* formula (8)) can also be rewritten as follows to be a standard form of *Least Square Loss Problem* with a $l1$-norm regularizer.

$$\Psi = \min_{\beta} \quad \|\mathcal{X}_2 \beta - \mathbf{y}_2\|_2^2 + \xi\|\beta\|_1 \tag{13}$$

where

$$\mathcal{X}_2 = \left[\mathbf{W}'\widehat{\mathbf{R}}\right], \quad \mathbf{y}_2 = \left[\mathbf{W}'\mathbf{r}\right]$$

Then the gradient *w.r.t* β can be derived as follows.

$$\frac{\partial \Psi}{\partial \beta} = 2\mathcal{X}_2^T \mathcal{X}_2 \beta - 2\mathcal{X}_2^T \mathbf{y}_2 + \xi I(\beta) \tag{14}$$

where $I(\beta)$ is an indicator function for all entries of β, defined in the same way as $I(\alpha)$ in formula (12). For details of the derivation, one can refer to A.2. And the SLEP package can also be utilized for optimizing Ψ and obtaining the optimal β for each tag in the vocabulary.

In the objective functions of the image-view and the tag-view linear sparse reconstructions, sometimes the to-be-reconstructed vectors (*i.e.* $\mathbf{f}, \mathbf{t}$ or $\mathbf{r}$) can be high-dimensional and the dictionary matrices (*i.e.* $\mathbf{F}, \widehat{\mathbf{T}}$ or $\widehat{\mathbf{R}}$) can be large. Then the computational cost of DLSR can be high. Here we propose that dimensionality reduction methods or sampling strategies like kNN (*i.e.* k Nearest Neighbors) can be adopted in that case for shrinking vectors or building smaller dictionary matrices while keeping acceptable performance.

4. Experiments

4.1. Experimental settings

To evaluate the proposed DLSR, we conduct extensive experiments on two widely-used benchmark datasets, *i.e.* Corel5k and IAPR TC12, and two web image datasets, *i.e.* NUS-WIDE and a new-built one named Flickr30Concepts. Some statistics of the four datasets are given in Table 1. With accurate manual annotations, the labeled tags of each image in Corel5k and IAPR TC12 are generally complete and contain few noises. Yet the vocabularies of both datasets are relatively small. And thus we further evaluate the proposed method on two much larger real-world web image datasets. The first one is the public NUS-WIDE dataset built by Chua et al.[37] with images randomly collected from Flickr. Following experiments in [27], we keep the top 1000 most frequent tags as its vocabulary for reducing random noises. Moreover, we build a new dataset named Flickr30Concepts by collecting images in a different way from NUS-WIDE. Specifically, we submit 30 non-abstract concepts[1] as queries to Flickr and collect the top 1000 of the retrieved images for each. Since the queries mostly correspond to small categories, the retrieved images for each are generally semantically related. We utilize WordNet for stemming and filtering the raw tags, and finally obtain a vocabulary containing 2513 distinct tags. Without any further reduction, this vocabulary could be more challenging than others ever used in experiments of previous related work [23,24,27,28]. Flickr30Concepts is also publicly available for research.

Table 1
Statistics of the benchmark Corel5k, IAPR TC12 and the real-world NUS-WIDE, Flickr30Concepts. Counts of tags are given in the format “mean/maximum”.

	Corel5k	IAPR TC12	NUS-WIDE	Flickr30Concepts
Vocabulary size	260	291	1000	2513
Nr. of images	4918	19,062	237,131	27,838
Tags per image	3.4/5	5.9/23	6.5/131	8.3/70
Del. tags per image	1.4 (40%)	2.3 (40%)	2.6 (40%)	3.3 (40%)
Test set	492	1,898	23,713	2,807

To perform tag completion, we randomly delete 40% of the labeled tags for all images in each dataset, and ensure that each image has at least one tag deleted and finally has at least one tag left. Therefore, we strike out images that are originally associated with only one tag. Finally we obtain four pretreated datasets with statistics shown in Table 1. Here all images in any dataset are partially labeled, which can be seen as the most challenging case of tag completion since no completely labeled images are provided. Then each dataset is split into test set (around 1/10) and training set. Note that we use the standard splits of the benchmark Corel5k and IAPR TC12 for experiments, as [5,7]. Moreover, we take around 1/9 of each training set as a validate set for parameter tuning. Due to the high costs of manual judgements for the tag completion results, in our experiments we take the deleted tags of each image as the ground-truth for measuring the performance of a tag completion method.

The experimental results of tag completion are measured with *average precision@N* (*AP@N*), *average recall@N* (*AR@N*) and *coverage@N* (*C@N*). For the top *N* tags added to a test image, *precision@N* is to measure the proportion of correct tags within the added ones, and *recall@N* is to measure the proportion of the ground-truth missing tags that are added, which are both averaged over all test images. *Coverage@N* is to measure the proportion of test images with at least one correctly added tag. All these performance metrics are respectively defined as follows.

$$AP@N = \frac{1}{m}\sum_{i=1}^{m}\frac{N_c(i)}{N} \tag{15}$$

$$AR@N = \frac{1}{m}\sum_{i=1}^{m}\frac{N_c(i)}{N_g} \tag{16}$$

$$C@N = \frac{1}{m}\sum_{i=1}^{m} I(N_c(i) > 0) \tag{17}$$

[1] The 30 non-abstract concepts are: aircraft, ball, beach, bike, bird, book, bridge, car, chair, child, clock, countryside, dog, door, fire, fish, flower, house, kite, lamp, mountain, mushroom, pen, rabbit, river, sky, sun, tower, train, tree.

where m is the number of test images, $N_c(i)$ is the number of correctly added tags to the ith image, $N_g(i)$ is the number of the ground-truth missing tags (*i.e.* the deleted ones) expected to be added to the ith image, and $I(\cdot)$ is a condition function that returns 1 when the condition is satisfied and 0 otherwise.

In our experiments, for Corel5k, IAPR TC12 and Flickr30Concepts, we utilize the open-source Lire project [38] for extracting ten kinds of low-level features[2] for each image, including global and local features, color and texture features, *etc.* And for NUS-WIDE, we utilize the provided six kinds of features.[3] Then for each dataset, we utilize principal component analysis (*i.e.* PCA) to separately perform dimensionality reduction for all features of an image, which are then concatenated to be a 400-dimensional merged feature vector. To measure the visual distance between images, we empirically utilize Euclidean distance for Edge Histogram, FCTH and Wavelet Texture, χ^2 distance for Color Layout and JCD, and Manhattan distance for the remaining features. Following JEC [5], distances of all features are normalized and combined with equal weights to be a final visual distance.

As mentioned previously, the proposed DLSR can be used as either an inductive method to perform tag completion for an unseen image or a transductive one for completing an existing dataset. For completing an unseen image, only images in the training set are utilized to build dictionary matrices. And for completing an existing dataset, DLSR can utilize all images in both training and test sets for linear sparse reconstructions, since all the to-be-completed images are already observed and also partially labeled, which can probably provide extra helpful information, and in this case we denote it as πDLSR, the transductive version of DLSR.

4.2. Tag completion results

In our experiments of tag completion, we adopt remarkable image auto-annotation methods (*i.e.* JEC [5] and TagProp [7]), tag recommendation approaches (*i.e.* Vote+ [2] and Folksonomy [13]) and recently proposed unified tag refinement frameworks of denoising and completion (*i.e.* LR [23] and SUG [24]) as baselines for making comparisons with the proposed DLSR. On each dataset, the model parameters of these baselines are carefully tuned on the corresponding validate set with their proposed tuning strategies for achieving their optimal performance under the same experimental settings, *e.g.* on Corel5k kNN = 200 for JEC and σML of TagProp, $[m, k_s, k_d, k_r] = [35, 3, 4, 2]$ for Vote+, *etc.* Actually the tuned parameter settings prove to yield much better performance than just using the published ones. And thus fairer comparisons can be further made with the proposed method.

For DLSR, to get more inside analyses of its reasonableness and effectiveness, we evaluate several of its variants. DLSR-IF and DLSR-IT are two variants that respectively perform tag completion via image-view linear sparse reconstruction with only low-level image features or initial tagging vectors, *i.e.* formula (2) and (3) with the proposed regularizers. Moreover, DLSR-I and DLSR-T are two variants that respectively perform tag completion via only image-view or tag-view linear sparse reconstruction, *i.e.* formula (6) and (8). Essentially DLSR-I is the combination of DLSR-IF and DLSR-IT. And DLSR is the proposed integral tag completion method combining both DLSR-I and DLSR-T. Meanwhile, πDLSR is the transductive version to complete all test images together, which utilizes images in both training and test sets to build dictionary matrices for linear sparse reconstructions. Note that in our experiments, the completion results of all methods are measured on the same test sets. For DLSR and its variants, to reduce the computational cost of image-view linear sparse reconstruction, we adopt a kNN strategy and take the 200 nearest visual neighbors of each test image to build the dictionary matrices. The model parameters of DLSR are carefully tuned on the validate set of Corel5k. Specifically, for μ, ω, λ and ξ in the image-view and the tag-view linear sparse reconstructions, *i.e.* formula (6) and (8), they are tuned via grid search with the corresponding value varying in $\{0, 2^{-3}, 2^{-2}, \ldots, 2^2, 2^3\}$. And for δ in the combination of image-view and tag-view reconstructed tagging vectors, *i.e.* formula (10), it is also tuned via grid search in $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$. According to the tag completion performance on the validate set, the optimal settings for $\mu, \omega, \lambda, \xi$ and δ are respectively $2^{-2}, 2, 1, 2^2$ and 0.5. Then we utilize them on test sets of Corel5k, IAPR TC12 and Flickr30Concetps in the subsequent experiments to validate the effectiveness of DLSR, and see how well the selected parameters on Corel5k can be generalized to the other datasets, since DLSR has more model parameters to tune than several baselines. Note that though we do not perform parameter tuning as other baselines for the proposed DLSR on the validate sets of IAPR TC12 and Flickr30Concepts, we perform parameter analyses for DLSR on all datasets to see the effects of model parameters and the differences between the selected parameter settings and the optimal ones on each dataset, as will be presented in the following subsection. For NUS-WIDE, as it provides totally different image features, we still need to perform parameter selection for DLSR on it. The corresponding selected optimal $\mu, \omega, \lambda, \xi$ and δ are respectively $2^{-2}, 2, 2^{-2}, 2^2$ and 0.3.

Table 2 presents the tag completion results of the proposed DLSR and its variants on the benchmark Corel5k and IAPR TC12, together with those of remarkable baselines. The experimental results are measured with $AP@N, AR@N$ and $C@N$, with N being 2 on Corel5k and 3 on IAPR TC12, since the average number of deleted tags per image on both datasets is respectively 1.4 and 2.3, as shown in Table 1. From the experimental results we can draw the following conclusions. (1) The proposed DLSR and its variants generally outperform the remarkable baselines, including image auto-annotation methods, tag recommendation approaches and unified tag refinement frameworks, which well demonstrates their effectiveness. (2) The variant DLSR-I outperforms both DLSR-IF and DLSR-IT, which well validates the necessity of concurrently considering both low-level image features and high-level initial tagging vectors in image-view linear sparse reconstruction.

Table 2
Tag completion results on the benchmark Corel5k and IAPR TC12, in terms of $AP@N, AR@N$ and $C@N$. Among the baselines, JEC and TagProp are image auto-annotation methods, Vote+ and Folksonomy are tag recommendation approaches, while LR and SUG are unified tag refinement frameworks of denoising and completion. Others are variants of the proposed DLSR. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k			IAPR TC12		
	$AP@2$	$AR@2$	$C@2$	$AP@3$	$AR@3$	$C@3$
JEC	0.23	0.33	0.39	0.20	0.26	0.44
TagProp	**0.27**	**0.40**	**0.48**	0.22	0.29	0.51
Vote+	0.25	0.37	0.45	0.20	0.26	0.48
Folksonomy	0.20	0.30	0.36	0.17	0.22	0.42
LR	**0.27**	**0.40**	0.47	**0.24**	**0.31**	**0.52**
SUG	0.25	0.38	0.45	0.20	0.26	0.48
DLSR-IF	0.28	0.41	0.49	0.23	0.31	0.53
DLSR-IT	0.26	0.37	0.45	0.23	0.30	0.50
DLSR-I	0.33	0.48	0.58	0.29	0.38	0.62
DLSR-T	0.28	0.41	0.49	0.22	0.30	0.53
DLSR	**0.34**	**0.50**	**0.59**	**0.30**	**0.41**	**0.65**
πDLSR	**0.34**	**0.50**	**0.59**	**0.31**	**0.42**	**0.66**

[2] The ten kinds of features include: Color Correlogram, Color Layout, CEDD, Edge Histogram, FCTH, JCD, Jpeg Coefficient Histogram, RGB Color Histogram, Scalable Color, SURF with Bag-of-Words model.

[3] The provided features include: Color Histogram, Color Correlogram, Edge Histogram, Wavelet Texture, Color Moments and SIFT with Bag-of-Words model.

Table 3
Tag completion results on the real-world NUS-WIDE and Flickr30Concepts, in terms of *AP*@*N*, *AR*@*N* and *C*@*N*. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	NUS-WIDE			Flickr30Concepts		
	AP@3	*AR*@3	*C*@3	*AP*@4	*AR*@4	*C*@4
JEC	0.06	0.07	0.14	0.25	0.30	0.49
TagProp	0.08	0.10	0.21	0.23	0.29	0.50
Vote+	**0.12**	**0.14**	**0.31**	0.23	0.27	0.48
Folksonomy	0.07	0.08	0.19	0.21	0.26	0.47
LR	–	–	–	**0.27**	**0.34**	**0.51**
SUG	–	–	–	–	–	–
DLSR-IF	0.07	0.08	0.18	0.21	0.30	0.48
DLSR-IT	0.09	0.10	0.20	0.26	0.28	0.42
DLSR-I	0.11	0.12	0.24	0.35	0.44	0.64
DLSR-T	0.16	0.20	0.39	0.30	0.36	0.60
DLSR	**0.18**	**0.22**	**0.42**	**0.38**	**0.48**	**0.71**
πDLSR	**0.18**	**0.22**	**0.42**	**0.39**	**0.48**	**0.72**

(3) DLSR outperforms both DLSR-I and DLSR-T, providing a strong evidence for the effectiveness and reasonableness of performing tag completion from both views of image and tag. (4) The transductive πDLSR slightly outperforms the inductive DLSR, since the former utilizes extra partially labeled images in the test set for building dictionary matrices while the latter does not.

Furthermore, we conduct experiments for all methods on the larger and more challenging real-world datasets, NUS-WIDE and Flickr30Concepts. Table 3 presents all the experimental results, in terms of *AP*@*N*, *AR*@*N* and *C*@*N*, with *N* being 3 on NUS-WIDE and 4 on Flickr30Concepts, as the average number of deleted tags per image on both datasets is respectively 2.6 and 3.3. Note that here we cannot obtain the tag completion results of SUG on both datasets due to its high computational cost to calculate the eigenvalues of the normalized Laplacian matrix of a large hyper-graph. We also cannot obtain the results of LR on NUS-WIDE due to its costly singular value decomposition for a large intermediate matrix with the same size as the initial tagging matrix. From Table 3 we can draw nearly the same conclusions as those on the benchmark Corel5k and IAPR TC12, which further demonstrate the effectiveness of the proposed DLSR. Moreover, since experiments of DLSR on both IAPR TC12 and Flickr30Concepts utilize the same parameter settings as those on Corel5k, the superior experimental results on both datasets in some sense demonstrate its robustness.

To compare with the most recently published tag completion methods, *i.e.* TMC [27] and DLC [28], we further conduct experiments on all datasets with new image features, because DLC requires the feature vector of each image to be non-negative and TMC prefers the dot product of feature vectors to being non-negative. Therefore, we utilize the SIFT feature with Bag-of-Words (BoW) model to represent each image, which is natively non-negative while many others are not. Moreover, SIFT is the only common feature used in the experiments of both methods and also the main feature utilized by TMC. Table 4 presents the experimental results of TMC, DLC and variants of DLSR on all datasets, which further demonstrate the effectiveness and reasonableness of the proposed method. Note that we cannot obtain the results of DLC on NUS-WIDE since the needed image-image correlation matrix is too large. It can be seen that both baselines yield inferior performance here, especially DLC. We attribute that to the following reasons: (1) both TMC and DLC are non-convex and may converge to a local optimum, (2) DLC depends heavily on image features for matrix factorization and calculating image similarities. For the proposed DLSR, with only SIFT feature, many retrieved visual neighbors of test images for building dictionary matrices are semantically unrelated, leading to a substantial performance degradation of the image-view linear sparse reconstruction. Yet boosted by the tag-view linear sparse reconstruction, the combined tag completion results (*i.e.* DLSR and πDLSR) are still acceptable and superior to those of the baselines, which well demonstrates the robustness of DLSR and the necessity of performing tag completion from both views of image and tag.

Moreover, to evaluate the model enhancements introduced in this paper, *i.e.* the diversity regularizer (formula (5)) and the ZMUV normalization method for combining image-view and tag-view results (formula (10)), we further compare the experimental results of DLSR with those of its previous version presented in [3]. Following [3], here the previous version is denoted as LSR and its transductive variant as πLSR. Table 5 presents the comparisons between the experimental results of DLSR and LSR, with those between πDLSR and πLSR, on all datasets with the various kinds of low-level image features. From the comparisons, we can observe that the introduced model enhancements in this paper can indeed help to gain performance improvement on most datasets, which demonstrates their effectiveness. For getting more inside details, we compare DLSR-I and its counterpart in [3], denoted as LSR-I, to validate the effectiveness of the introduced diversity regularizer in the image-view linear sparse reconstruction, as shown in Table 6. It can be seen that on all datasets, the introduced diversity regularizer can help to improve the performance of image-view reconstruction, which demonstrates the reasonableness of considering diverse tag information for tag completion. Moreover, we compare the combination method in this paper and that used in [3] to validate the effectiveness of the former, with the latter denoted as DLSR*, as shown in Table 7. From the comparisons, we can observe that the new combination method with ZMUV normalization in this paper generally yields slightly better performance on most datasets. To conclude, experimental results show that the model enhancements introduced in this paper are reasonable and generally can help to gain performance improvement over the previous version presented in [3].

Table 4
Tag completion results of TMC, DLC and variants of the proposed DLSR on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with only SIFT BoW feature, in terms of *AP*@*N*, *AR*@*N* and *C*@*N*. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k ($N = 2$)			IAPR TC12 ($N = 3$)			NUS-WIDE ($N = 3$)			Flickr30Concepts ($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
TMC	**0.23**	**0.33**	**0.40**	**0.14**	**0.20**	**0.37**	**0.13**	**0.15**	**0.32**	**0.19**	**0.21**	**0.37**
DLC	0.09	0.13	0.18	0.10	0.12	0.27	–	–	–	0.07	0.09	0.23
DLSR-IF	0.13	0.18	0.24	0.08	0.10	0.20	0.03	0.03	0.07	0.06	0.08	0.16
DLSR-IT	0.15	0.23	0.27	0.10	0.13	0.24	0.05	0.05	0.12	0.09	0.09	0.17
DLSR-I	0.19	0.28	0.34	0.13	0.17	0.31	0.05	0.06	0.13	0.12	0.14	0.25
DLSR-T	0.28	0.41	0.49	0.22	0.30	0.53	0.16	0.20	0.39	0.30	0.36	0.60
DLSR	**0.28**	**0.42**	**0.50**	**0.23**	**0.31**	**0.55**	**0.17**	**0.20**	**0.40**	**0.31**	**0.37**	**0.61**
πDLSR	**0.29**	**0.43**	**0.51**	**0.23**	**0.31**	**0.55**	**0.17**	**0.20**	**0.40**	**0.32**	**0.38**	**0.62**

Table 5
Comparisons between the experimental results of DLSR and its previous version LSR presented in [3] on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the various kinds of low-level image features, in terms of $AP@N, AR@N$ and $C@N$. Comparisons between their corresponding transductive variants, *i.e.* πDLSR and πLSR, are also presented. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k ($N = 2$)			IAPR TC12 ($N = 3$)			NUS-WIDE ($N = 3$)			Flickr30Concepts ($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
LSR	0.33	0.48	0.58	**0.30**	**0.41**	0.64	**0.18**	**0.22**	**0.42**	0.37	0.45	0.67
DLSR	**0.34**	**0.50**	**0.59**	**0.30**	**0.41**	**0.65**	**0.18**	**0.22**	**0.42**	**0.38**	**0.48**	**0.71**
πLSR	0.33	0.49	0.58	**0.31**	0.41	0.65	**0.18**	**0.22**	**0.42**	0.38	0.46	0.69
πDLSR	**0.34**	**0.50**	**0.59**	**0.31**	**0.42**	**0.66**	**0.18**	**0.22**	**0.42**	**0.39**	**0.48**	**0.72**

Table 6
Comparisons between DLSR-I and its counterpart in [3], denoted as LSR-I, on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the various kinds of low-level image features, in terms of $AP@N, AR@N$ and $C@N$. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k ($N = 2$)			IAPR TC12 ($N = 3$)			NUS-WIDE ($N = 3$)			Flickr30Concepts ($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
LSR-I	0.30	0.45	0.54	0.28	**0.38**	0.60	0.10	0.11	0.23	0.33	0.40	0.60
DLSR-I	**0.33**	**0.48**	**0.58**	**0.29**	**0.38**	**0.62**	**0.11**	**0.12**	**0.24**	**0.35**	**0.44**	**0.64**

Table 7
Comparisons between the combination method in this paper (*i.e.* DLSR) and that used in [3] (*i.e.* DLSR*), on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the various kinds of low-level image features, in terms of $AP@N, AR@N$ and $C@N$. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k ($N = 2$)			IAPR TC12 ($N = 3$)			NUS-WIDE ($N = 3$)			Flickr30Concepts ($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
DLSR*	**0.35**	**0.51**	**0.60**	**0.30**	0.40	**0.65**	**0.18**	0.21	**0.42**	0.37	0.47	0.70
DLSR	0.34	0.50	0.59	**0.30**	**0.41**	**0.65**	**0.18**	**0.22**	**0.42**	**0.38**	**0.48**	**0.71**

4.3. Parameter analyses

To evaluate the effects of model parameters in the proposed DLSR, *i.e.* μ, λ, ω in the image-view linear sparse reconstruction (*i.e.* formula (6)) and ξ in the tag-view linear sparse reconstruction (*i.e.* formula (8)), we use the control variable method to perform parameter analyses on the test sets of all datasets. Specifically, we use the same parameter settings as former experiments of tag completion, and then vary the value of a parameter in $\{0, 2^{-3}, 2^{-2}, \ldots, 2^{2}, 2^{3}\}$ with others fixed to see the performance variations.

Fig. 4 presents the experimental results of parameter analyses on the benchmark Corel5k, in terms of $AP@2, AR@2$ and $C@2$. It can be seen that the optimal parameter settings for μ, λ, ω and ξ (sub-Fig. 4(a)–(d)) are respectively close to the ones that we select on the validate set of Corel5k for the former experiments of tag completion. And we can also see that for each parameter, all the performance curves *w.r.t* precision, recall and coverage are convex on the whole, which reflects the significance of the corresponding part included in the objective functions. Particularly, we find that the effects of ω and ξ are much less significant than those of μ and λ. We attribute it to the following reasons: (1) for ω, it is expected to work when the reconstruction weights are dominating in only images containing the same initial tagging vector as the to-be-completed image, which in fact happens occasionally, (2) for ξ, as a tag generally co-occurs with only a few semantically related ones, the obtained reconstruction weights are in some way naturally sparse, and thus ξ is expected to work for tags that co-occur with many others.

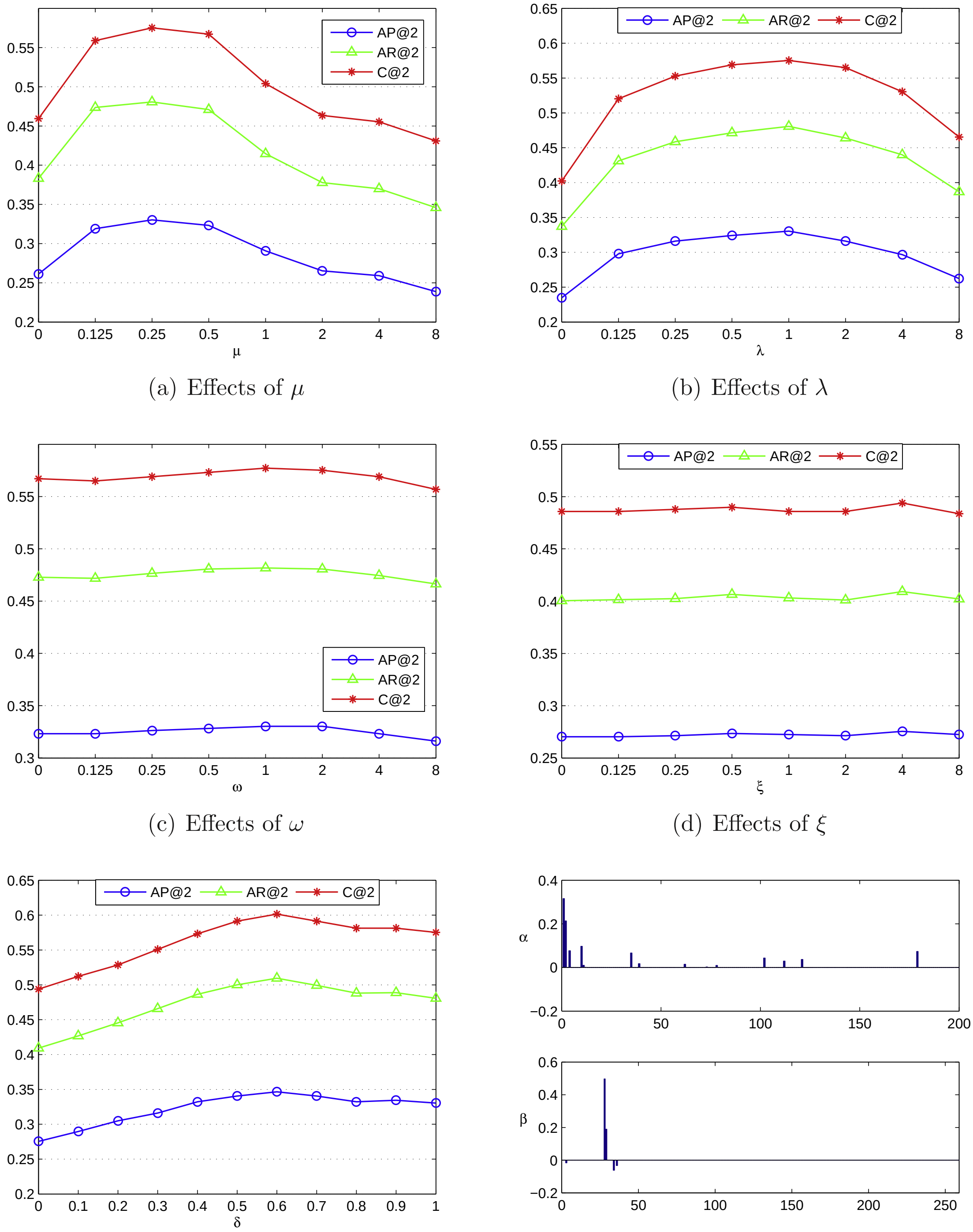
We further investigate the effects of the weighting parameter δ for the combination of image-view and tag-view reconstructed tagging vectors (*i.e.* formula (10)), by varying δ from 0 to 1 with a step of 0.1. As shown in Fig. 4(e), the optimal setting for δ is around 0.6, which is close to the one selected for former experiments of tag completion (*i.e.* 0.5). Moreover, it can be observed that the optimal combined tag completion result outperforms that of single image-view (*i.e.* $\delta = 1$) or single tag-view (*i.e.* $\delta = 0$) linear sparse reconstruction, which well demonstrates the effectiveness and reasonableness of combining both the image-view and the tag-view results.

Furthermore, to provide inside details of the optimization for both the image-view and the tag-view objective functions, here we give a sample of optimized α and β, as shown in Fig. 4(f). It is obvious that the optimized α and β are both sparse, as is expected by the corresponding introduced constraints of sparsity.

Similar experimental results and conclusions can also be observed on IAPR TC12, NUS-WIDE and Flickr30Concepts. Specifically, we find that the optimal parameter settings on IAPR TC12 and Flickr30Concepts are also close to the ones selected on the validate set of Corel5k, meaning that the selected parameters on Corel5k can be well generalized to these two datasets with the same kinds of image features. And thus parameter settings of the proposed DLSR seem to be less dependent on datasets. Moreover, the optimal parameter settings on NUS-WIDE, which provides different image features, are also mostly close to the ones that we select on the corresponding validate set. For more details, one can refer to Appendix B.

4.4. Tag completion with noisy initial tags

As elaborated previously, for exploiting image-image similarities, the proposed DLSR reconstructs the initial tagging vector of any to-be-completed image with those of others in the image-view linear sparse reconstruction, and for discovering tag-tag correlations, DLSR reconstructs the initial tagging column vector of each tag with those of others in the tag-view linear sparse



(a) Effects of μ

(b) Effects of λ

(c) Effects of ω

(d) Effects of ξ

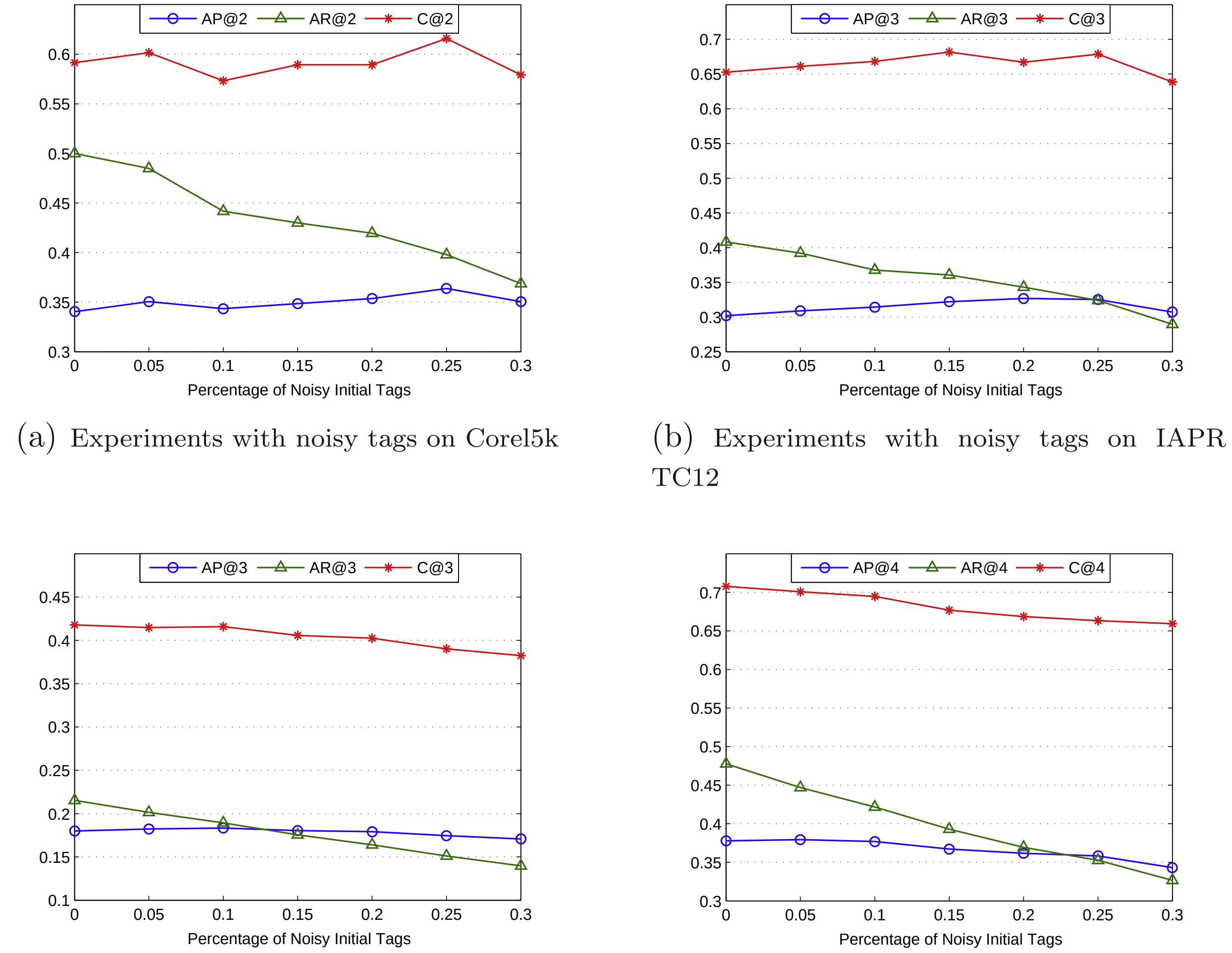
(e) Effects of δ

(f) A sample of optimized α and β

Fig. 4. Effects of μ, λ, ω in the objective function of image-view linear sparse reconstruction (sub-Fig. 4(a)–(c)), ξ in the objective function of tag-view linear sparse reconstruction (sub-Fig. 4(d)), and δ for combining the image-view and the tag-view reconstructed tagging vectors (sub-Fig. 4(e)), in terms of $AP@2$, $AR@2$ and $C@2$ on the test set of the benchmark Corel5k, with sub-Fig. 4(f) giving a sample of optimized α (upper) and β (lower).

reconstruction. Then it would be interesting to investigate how DLSR can tolerate with noisy initial tags.

Specifically, in our experiments, we randomly delete different percentages (*i.e.* $[0\%, 5\%, 10\%, \ldots, 30\%]$) of the given initial tags in both training set and test set, and replace them with noisy ones, *i.e.* semantically unrelated tags. Then we evaluate the proposed DLSR in those cases with different percentages of noisy initial tags, as illustrated in Fig. 5. From the experimental results, we can observe that as the percentage of noisy initial tags increases from 0% to 30%, the performance of DLSR tends to decrease on the whole, with the *recall* decreasing significantly while the *precision* and *coverage* varying a little. The significant decrease of *recall* is due to that a higher percentage of noisy tags corresponds to more ground-truth missing tags and thus larger denominators in formula (16), resulting in a substantially lower *recall*. Another interesting observation is that with noisy initial tags, DLSR can sometimes achieve slightly better *precision* and *coverage*. It is because that more missing tags offer DLSR more chances to get related tags from the unlabeled candidates. To conclude, in terms of *precision* and *coverage*, the proposed DLSR has certain tolerance



(a) Experiments with noisy tags on Corel5k

(b) Experiments with noisy tags on IAPR TC12

(c) Experiments with noisy tags on NUS-WIDE

(d) Experiments with noisy tags on Flickr30Concepts

Fig. 5. Performance variations of the proposed DLSR with different percentages of noisy initial tags on Corel5k (sub-Fig. 5(a)), IAPR TC12 (sub-Fig. 5(b)), NUS-WIDE (sub-Fig. 5) and Flickr30Concepts (sub-Fig. 5(d)), in terms of *AP*@*N*, *AR*@*N* and *C*@*N*.

for noisy initial tags, which can be attributed to the sparsity regularizers introduced in the dual-view reconstructions.

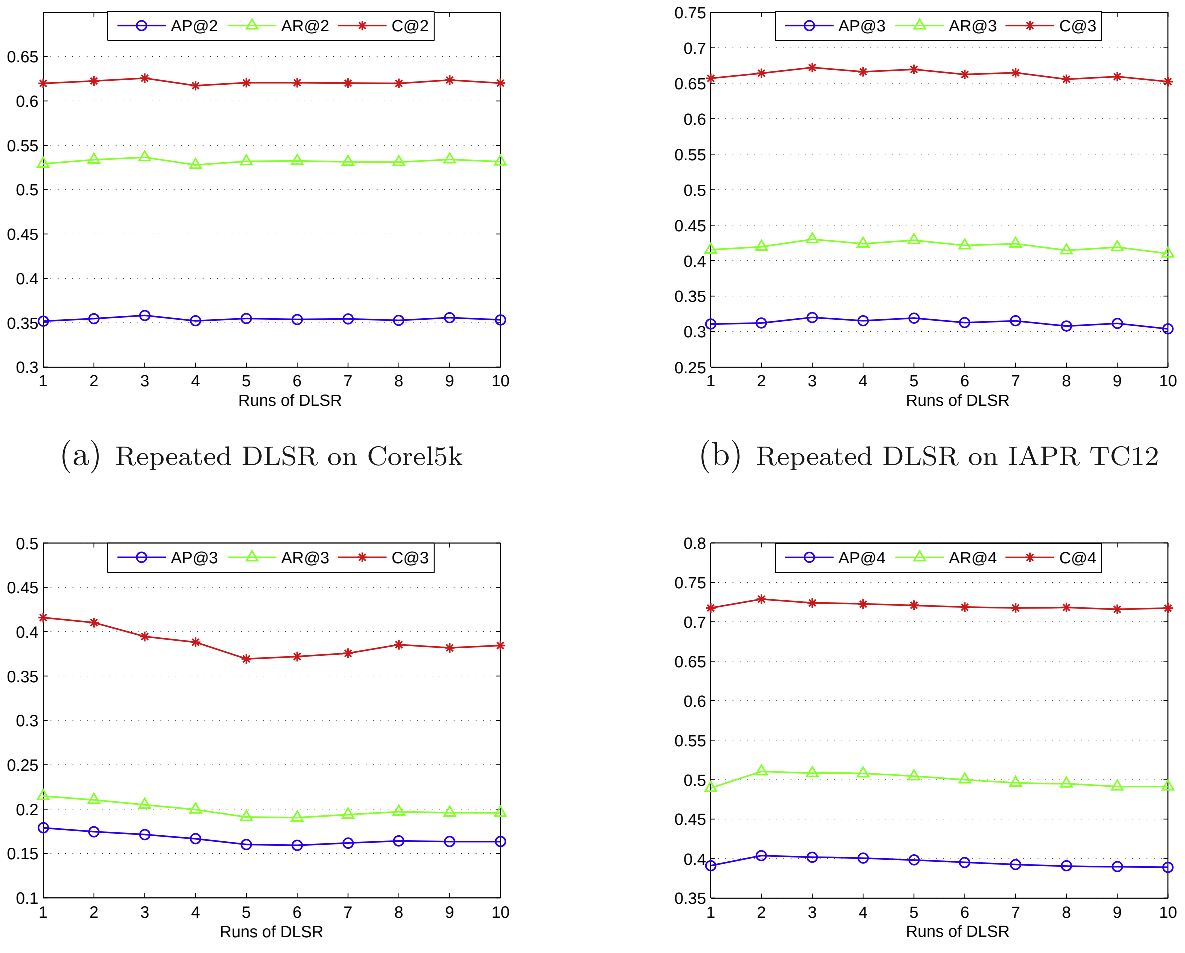
4.5. Tag completion with repeated DLSR

Different from previous related work on tag completion [27,28] that performs global refinement for the initial tagging matrix, the proposed DLSR performs tag completion via linearly reconstructing each image and each tag separately. As DLSR is not a global refinement approach, when it is used to perform tag completion for an existing dataset, one may want to see whether running DLSR over and over can help to obtain better tag completion results. That is, after performing tag completion for all images in the dataset with DLSR, the completed tagging matrix can be further used to run DLSR again for all images to get a new completed tagging matrix. In a similar fashion, we can perform DLSR repeatedly for the to-be-completed dataset, and see whether it can finally yield better tag completion results than performing DLSR only once.

Specifically, when performing DLSR repeatedly, the initial tagging matrix will be the input of the 1st run of DLSR. And then for each run, its output, *i.e.* the completed tagging matrix, will be used as the input of the next run. In our experiments, to keep as much information as possible, we use the real-valued tag completion result of a run of DLSR as the input of the next one. Moreover, we linearly normalize the tagging vector of each image into $[0, 1]$ to prevent numerical divergence, and recover the corresponding values of initially labeled tags as 1. For each run, we measure its tag completion result with *AP*@*N*, *AR*@*N* and *C*@*N* on the same test sets as previous experiments, in order to get more inside details. Fig. 6 presents the performance variations of performing DLSR repeatedly in 10 runs on Corel5k ($N = 2$), IAPR TC12 ($N = 3$), NUS-WIDE ($N = 3$) and Flickr30Concepts ($N = 4$). Note that for the 1st run on each dataset, the presented performance corresponds to the tag completion result of DLSR, not the quality of the initial tagging matrix. It is evident that on all datasets, the tag completion performance generally varies little with runs of DLSR. Though on some datasets like IAPR TC12, a few more runs may gain slight performance improvement, too many runs can still result in performance degradations due to the accumulative noises of previous ones. Therefore, it can be concluded that performing DLSR once can generally be effective enough for tag completion, and there is no need to perform it repeatedly, since more runs generally cannot gain substantial performance improvement and can even lead to performance degradations.

4.6. Further evaluation with a completely labeled training set

To further evaluate the proposed DLSR, we conduct experiments on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the corresponding training set completely labeled, in order



(a) Repeated DLSR on Corel5k

(b) Repeated DLSR on IAPR TC12

(c) Repeated DLSR on NUS-WIDE

(d) Repeated DLSR on Corel5k

Fig. 6. Performance variations of performing the proposed DLSR repeatedly in 10 runs on Corel5k (sub-Fig. 6(a)), IAPR TC12 (sub-Fig. 6(b)), NUS-WIDE (sub-Fig. 6) and Flickr30Concepts (sub-Fig. 6(d)), in terms of *AP*@*N*, *AR*@*N* and *C*@*N*.

Table 8
Experimental results of tag completion on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the various kinds of low-level image features and completely labeled training sets, in terms of *AP*@*N*, *AR*@*N* and *C*@*N*. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k			IAPR TC12			NUS-WIDE			Flickr30Concepts		
	($N = 2$)			($N = 3$)			($N = 3$)			($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
JEC	0.28	0.41	0.47	0.25	0.33	0.50	0.07	0.08	0.15	0.35	0.44	0.54
TagProp	**0.32**	**0.47**	**0.54**	**0.29**	**0.38**	**0.60**	0.10	0.11	0.23	**0.36**	**0.46**	0.61
Vote+	0.27	0.40	0.49	0.21	0.27	0.50	**0.13**	**0.16**	**0.34**	0.31	0.39	**0.62**
Folksonomy	0.23	0.34	0.40	0.20	0.27	0.44	0.08	0.10	0.21	0.27	0.35	0.50
LR	0.30	0.44	0.51	0.27	0.35	0.57	–	–	–	0.31	0.40	0.55
SUG	0.27	0.40	0.48	0.23	0.31	0.53	–	–	–	–	–	–
DLSR	**0.36**	**0.53**	**0.62**	**0.34**	**0.47**	**0.71**	**0.20**	**0.25**	**0.46**	**0.44**	**0.57**	**0.76**
πDLSR	**0.37**	**0.55**	**0.65**	**0.35**	**0.48**	**0.72**	**0.20**	**0.25**	**0.46**	**0.47**	**0.61**	**0.80**

Table 9
Experimental results of tag completion on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with SIFT BoW feature and completely labeled training sets, in terms of *AP*@*N*, *AR*@*N* and *C*@*N*. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k			IAPR TC12			NUS-WIDE			Flickr30Concepts		
	($N = 2$)			($N = 3$)			($N = 3$)			($N = 4$)		
	AP	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*	*AP*	*AR*	*C*
TMC	**0.24**	**0.36**	**0.43**	**0.17**	**0.23**	**0.43**	**0.14**	**0.17**	**0.34**	**0.18**	**0.21**	**0.36**
DLC	0.10	0.14	0.20	0.10	0.13	0.28	–	–	–	0.06	0.10	0.23
DLSR	**0.28**	**0.43**	**0.52**	**0.24**	**0.33**	**0.56**	**0.17**	**0.21**	**0.42**	**0.34**	**0.45**	**0.70**
πDLSR	**0.29**	**0.45**	**0.54**	**0.24**	**0.33**	**0.56**	**0.17**	**0.21**	**0.42**	**0.34**	**0.45**	**0.70**

Table 10
Experimental results of tag recommendation by Vote+, Folksonomy and the proposed DLSR on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with each test image initially labeled with only one tag, in terms of P@1, P@5 and S@5. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

		P@1	P@5	S@5
Corel5k	Vote+	0.41	0.23	0.68
	Folksonomy	0.40	0.21	0.68
	DLSR	**0.56**	**0.29**	**0.84**
IAPR TC12	Vote+	0.40	0.24	0.71
	Folksonomy	0.38	0.26	0.69
	DLSR	**0.53**	**0.34**	**0.84**
NUS-WIDE	Vote+	0.24	0.14	0.46
	Folksonomy	0.19	0.12	0.38
	DLSR	**0.27**	**0.17**	**0.51**
Flickr30Concepts	Vote+	0.33	0.22	0.49
	Folksonomy	0.40	0.26	0.56
	DLSR	**0.47**	**0.30**	**0.71**

Table 11
Average time costs (in seconds) for Vote+, Folksonomy and the proposed DLSR to perform tag recommendation for an image in Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts. Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Corel5k	IAPR TC12	NUS-WIDE	Flickr30Concepts
Vote+	**0.0006**	**0.0001**	**0.0004**	**0.0004**
Folksonomy	0.0031	0.0085	0.0619	0.0203
DLSR	0.0334	0.0364	0.0490	0.0495

to see whether it can still yield superior performance in better cases of tag completion. Specifically, we recover all the deleted tags in the training sets of all datasets. Then DLSR and other baselines are applied to the same partially labeled test sets as former experiments.

Table 8 presents the experimental results of JEC [5], TagProp [7], Vote+ [2], Folksonomy [13], LR [23], SUG [24], DLSR and its transductive version πDLSR for tag completion on all datasets with the various kinds of low-level image features, in terms of $AP@N$, $AR@N$ and $C@N$. Moreover, Table 9 reports the experimental results of TMC [27], DLC [28], DLSR and πDLSR on all datasets with only SIFT BoW feature. Then we can draw the following conclusions. (1) The proposed DLSR and πDLSR still outperform other baselines on all datasets with completely labeled training sets, which further demonstrates their effectiveness. (2) All methods generally achieve performance improvement with a completely labeled training set, as can be seen by comparing with the experimental results in Table 2–4.

5. Applications

5.1. Tag recommendation

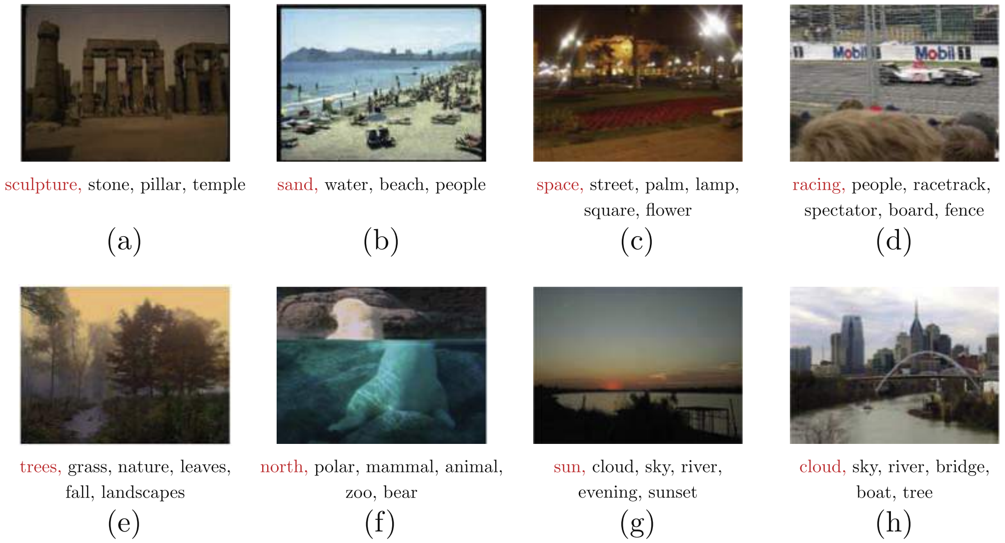
As demonstrated by the former experiments, the proposed DLSR achieves encouraging tag completion performance. Then if the time cost of DLSR is acceptable, it would be feasible to be used for online tag recommendation to help users to label their pictures with less efforts, as Vote+ [2] and Folksonomy [13] do. Actually, in the proposed DLSR, since the tag-view linear sparse reconstruction is to learn a tag-tag correlation matrix $\mathcal{B}$ that is shared by all unseen to-be-completed images, it can be performed and updated offline, and then used for obtaining tag-view reconstructed tagging vectors for to-be-completed images with formula (9). As for the image-view linear sparse reconstruction, to facilitate the optimization process, it would be preferable to firstly retrieve the kNN of a to-be-completed image and then utilize their corresponding image features and initial tagging vectors to build smaller dictionary matrices, as our former experiments do. Recently with the development of approximate kNN retrieval methods like locality sensitive hash [39–42], the time cost to retrieve kNN from millions of images could be only a few milliseconds, benefiting from the fast bit operations for calculating hamming-based visual distances between images. Moreover, with a kNN sampling strategy, the numbers of images and candidate tags in the kNN set are generally much smaller than the scale of the dataset, thus lowering the computational cost of the image-view reconstruction and making it less dependent on the scale of the dataset. Therefore, with the optimization strategies proposed above, the tag completion process of DLSR can be substantially accelerated.

To evaluate the proposed DLSR for tag recommendation, we further conduct experiments on all datasets. Note that we use the same test sets as former experiments of tag completion. And for better simulating the real-world labeling process, we randomly keep only one initial tag for each test image. That is, we evaluate the performance of DLSR and the baselines (*i.e.* Vote+ [2] and Folksonomy [13]) to recommend tags for a user after he has labeled one tag for a test image. Experimental results are measured with the widely-used performance metrics in the field of tag recommendation [2,43], *i.e. Precision at rank k* (P@k) and *Success at rank k* (S@k). Specifically, *Precision at rank k* is the average proportion of correct tags among the *top k* recommended ones, and *Success at rank k* is defined as the probability of finding a correct tag in the *top k* recommended ones. Table 10 presents the experimental results of tag recommendation by Vote+, Folksonomy and the proposed DLSR, in terms of P@1, P@5 and S@5. Note that S@1 is equal to P@1, and here it is left out for clarity. It can be concluded that the proposed DLSR significantly outperforms Vote+ and Folksonomy on all datasets. And thus it is effective for tag recommendation.

Here all experiments of tag recommendation are performed with Matlab 8.1 on a PC with an Intel Core i5-2400 CPU and 4G RAM. For the proposed DLSR, as shown in Table 11, with the tag-tag correlation matrix $\mathcal{B}$ in the tag-view linear sparse reconstructions calculated offline, the average time costs to recommend tags for a test image in Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concept are respectively 0.03 s, 0.04 s, 0.05 s and 0.05 s. The differences between time costs on different datasets can be attributed to both the different vocabulary sizes and the different average numbers of candidate tags appearing in the kNN set of a to-be-completed image. Particularly, for the large NUS-WIDE and Flickr30Concepts, by retrieving the kNN of a to-be-completed

Table 12
Experimental results of image auto-annotation by the state-of-the-art TagProp on Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concepts with the training sets differently labeled, in terms of *average tag-based precision* (**p**) and *average tag-based recall* (**r**). Numbers in bold highlight the best performance achieved by baselines or variants of the proposed DLSR.

	Training set	**p**	**r**
Corel5k	Incompletely labeled	0.28	0.28
	Completed by DLSR	**0.30**	**0.29**
	Perfectly labeled	0.30	0.33
IAPR TC12	Incompletely labeled	0.41	0.17
	Completed by DLSR	**0.46**	0.17
	Perfectly labeled	0.51	0.24
NUS-WIDE	Incompletely labeled	0.23	0.04
	Completed by DLSR	**0.26**	0.04
	Perfectly labeled	0.27	0.06
Flickr30Concepts	Incompletely labeled	0.26	0.14
	Completed by DLSR	**0.30**	**0.17**
	Perfectly labeled	0.39	0.24



sculpture, stone, pillar, temple

(a)

sand, water, beach, people

(b)

space, street, palm, lamp, square, flower

(c)

racing, people, racetrack, spectator, board, fence

(d)

trees, grass, nature, leaves, fall, landscapes

(e)

north, polar, mammal, animal, zoo, bear

(f)

sun, cloud, sky, river, evening, sunset

(g)

cloud, sky, river, bridge, boat, tree

(h)

Fig. 7. Samples of tag recommendation results by DLSR on Corel5k (*i.e.* sample (a)–(b)), IAPR TC12 (*i.e.* sample (c)–(d)), NUS-WIDE (*i.e.* sample (e)–(f)) and Flickr30Concepts (*i.e.* sample (g)–(h)), with the red tag being the only one user-provided tag and others being the recommended ones. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

image, the size of the image-view optimization problem is significantly reduced, as the size of a kNN set is much smaller than the total image number and the number of candidate tags appearing in it is also much smaller than the vocabulary size. From the experimental results on all datasets, it can be observed that though the time costs of DLSR are generally higher than the previous methods, they are still acceptable for online tag recommendation and DLSR can also yield superior performance.

Fig. 7 gives samples of tag recommendation results by the proposed DLSR on all datasets, with the red tag being the only one user-provided tag and others being the recommended ones. For Corel5k (*i.e.* sample (a)–(b)), as the mean number of tags per image is 3.4, we only recommend the top 3 tags. And for IAPR TC12 (*i.e.* sample (c)–(d)), NUS-WIDE (*i.e.* sample (e)–(f)) and Flickr30Concepts (*i.e.* sample (g)–(h)), we recommend the top 5 tags. From the samples, we can find that DLSR can well recommend semantically related tags for a given image, even with only one initially labeled tag.

5.2. Data pretreatment for image auto-annotation

As mentioned previously, incompleteness of user-provided tags in images can lead to performance degradations for various tag-dependent applications, *e.g.* image auto-annotation. Therefore, the proposed DLSR can be utilized as a data pretreatment method to perform tag completion for any incompletely labeled dataset.

Here we conduct experiments of image auto-annotation on all datasets to see whether the pretreatment by DLSR can help to achieve performance enhancements. We utilize the state-of-the-art image auto-annotation method, TagProp [7], for experiments, and use the standard dataset splits of the benchmark Corel5k and IAPR TC12 as previous work [5,7]. Here the training sets are the same as former experiments and thus incompletely labeled, with 40% of the labeled tags deleted. And all images in the test sets are totally unlabeled. Then for each dataset, we perform TagProp on the same test set with the training set differently labeled. That is, the training set can be the incompletely labeled one with 40% of the tags deleted, or the one completed by the proposed DLSR, or the perfectly labeled one which recovers all the deleted tags. Since the average number of deleted tags per training image in Corel5k, IAPR TC12, NUS-WIDE and Flickr30Concpets is respectively 1.4, 2.3, 2.6 and 3.3, as shown in Table 1, here we utilize DLSR to conservatively add another 1 tag for each training image in Corel5k, 2 tags for IAPR TC12 and NUS-WIDE, and 3 tags for Flickr30Concepts. Following previous work of image auto-annotation [4–7], each test image is annotated with the top 5 tags by TagProp. And experimental results are measured with widely-used performance metrics for image auto-annotation, *i.e. average tag-based precision*(**p**) and *average tag-based recall* (**r**). With the annotations predicted by TagProp, for each tag, the *tag-based precision* measures the proportion of retrieved images that are ground-truth related ones, and the *tag-based recall* is defined as the proportion of ground-truth related images that are retrieved. Both are averaged over all tags to be the *average tag-based precision* and *average tag-based recall*.

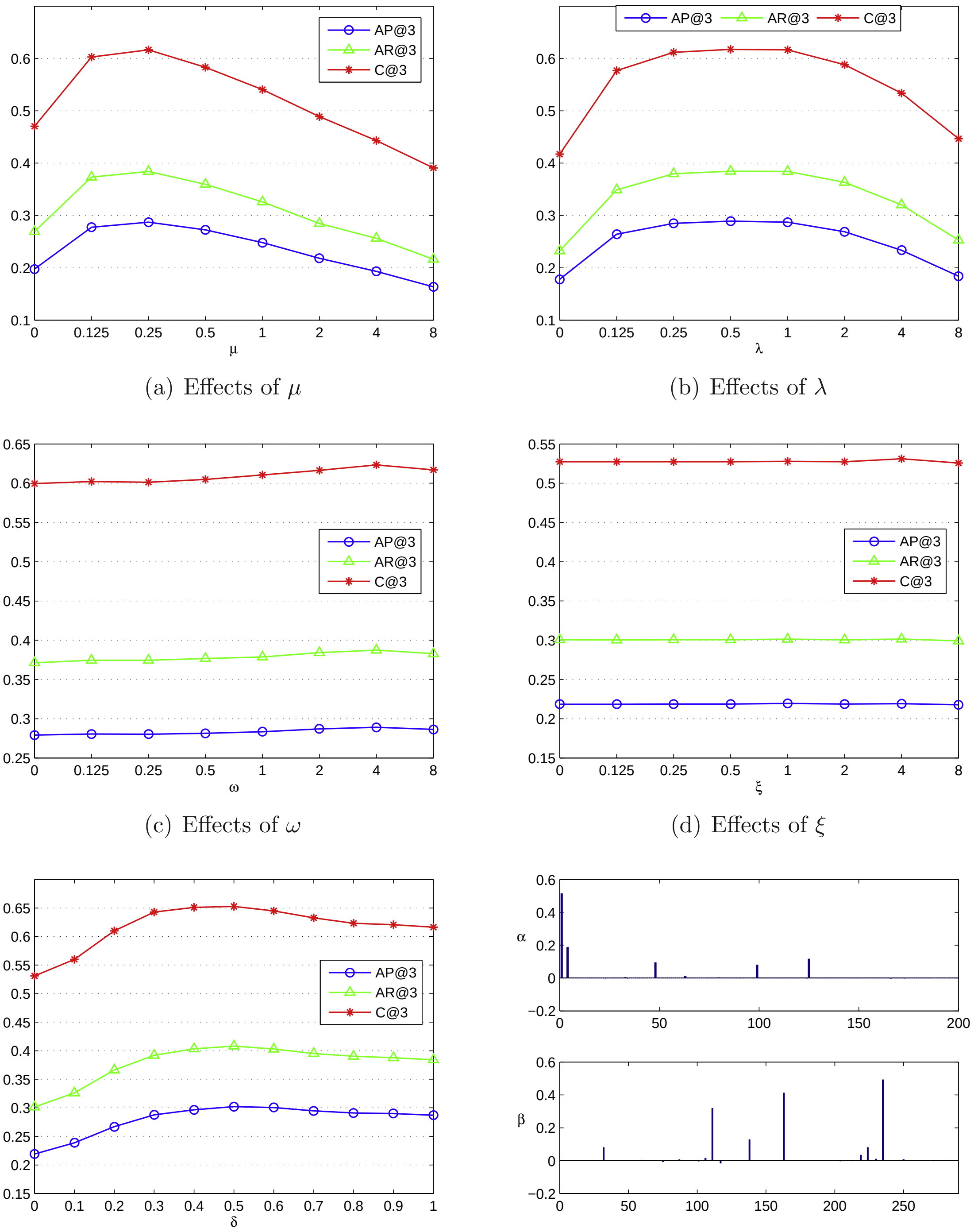
Table 12 presents the experimental results of TagProp on all datasets, with the corresponding training set differently labeled. Then we can draw the following conclusions. (1) With the proposed DLSR as a data pretreatment method to complete the incompletely labeled training set, TagProp can generally achieve performance enhancements on all datasets, well demonstrating the effectiveness of DLSR. (2) By comparing the experimental results of TagProp using the incompletely labeled training set with those using the perfectly labeled one, we can observe that incompleteness of tags in images can lead to significant performance degradations for image auto-annotation, which further validates the necessity of performing tag completion for incompletely labeled images. (3) Compared with the perfectly labeled training set, the one completed by the proposed DLSR is still inferior in promoting TagProp, meaning that the field of tag completion deserves more further researches.

6. Conclusions

In this paper we propose an effective method denoted as DLSR for automatic image tag completion via dual-view linear sparse reconstructions. Specifically, for any to-be-completed image, the image-view linear sparse reconstruction exploits the image-image correlations to obtain an image-view reconstructed tagging vector with those of others. And the tag-view linear sparse reconstruction exploits the tag-tag correlations to obtain a tag-view reconstructed tagging vector with the initially labeled tags. Then both are combined with effective normalization and combination strategies in the field of meta-search for better predicting missing related tags. In the proposed DLSR, both the image-view and the tag-view linear sparse reconstructions are respectively fit into convex optimization frameworks, considering various available contextual information. DLSR is evaluated with extensive experiments conducted on benchmark datasets and real-world web images. And experimental results well demonstrate that it is effective and reasonable. DLSR can also help to enhance a variety of tag-dependent applications.

Acknowledgments

This research was supported by the National Basic Research Project of China (Grant No. 2011CB70700), the National Natural Science Foundation of China (Grant No. 61271394, 61005045), and the National HeGaoJi Key Project (No. 2013ZX01039-002-002). And the authors would like to thank the reviewers for their valuable comments.



(a) Effects of μ

(b) Effects of λ

(c) Effects of ω

(d) Effects of ξ

(e) Effects of δ

(f) A sample of optimized α and β

Fig. B.8. Effects of μ, λ, ω in the objective function of image-view linear sparse reconstruction (sub-Fig. B.8(a)–(c)), ξ in the objective function of tag-view linear sparse reconstruction (sub-Fig. B.8(d)), and δ for combining the image-view and the tag-view reconstructed tagging vectors (sub-Fig. B.8(e)), in terms of $AP@3$, $AR@3$ and $C@3$ on the test set of the benchmark IAPR TC12, with sub-Fig. B.8(f) giving a sample of optimized α (upper) and β (lower).

Appendix A. Proofs for convexity of the image-view and the tag-view objective functions

A.1. Convexity of the image-view objective function

As pointed out in this paper, the objective function of the image-view linear sparse reconstruction, *i.e.* the following formula, can be demonstrated to be convex and thus there exists a global optimal solution.

$$\begin{aligned}\Theta = \min_{\alpha} \quad & \|\mathbf{f} - \mathbf{F}\alpha\|_2^2 + \mu\|\mathbf{W}\left(\mathbf{t} - \widehat{\mathbf{T}}\alpha\right)\|_2^2 + \omega\|\mathbf{s}^T\alpha\|_2^2 \\ & + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right)\end{aligned}$$

As the weighting parameters (*i.e.* μ, ω, λ) are non-negative, Θ can be further rewritten as follows to be a standard form of *Least Squares Loss Problem* [35] regularized by an overlapping sparse group lasso.

$$\Theta = \min_{\alpha} \quad \|\mathcal{X}_1\alpha - \mathbf{y}_1\|_2^2 + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right)$$

where

$$\mathcal{X}_1 = \begin{bmatrix}\mathbf{F} \\ \mathbf{W}\widehat{\mathbf{T}}\sqrt{\mu} \\ \mathbf{s}^T\sqrt{\omega}\end{bmatrix}, \qquad \mathbf{y}_1 = \begin{bmatrix}\mathbf{f} \\ \mathbf{W}\mathbf{t}\sqrt{\mu} \\ 0\end{bmatrix}$$

Then we can derive the following formula.

$$\begin{aligned}\Theta \quad & = \min_{\alpha} \quad \|\mathcal{X}_1\alpha - \mathbf{y}_1\|_2^2 + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right) \\ & = \min_{\alpha} \quad (\mathcal{X}_1\alpha - \mathbf{y}_1)^T(\mathcal{X}_1\alpha - \mathbf{y}_1) + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right) \\ & = \min_{\alpha} \quad \alpha^T\mathcal{X}_1^T\mathcal{X}_1\alpha - 2\mathbf{y}_1^T\mathcal{X}_1\alpha + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\|g_i\|_2\right) + \mathbb{C}_1\end{aligned}$$

where $\mathbb{C}_1$ is a constant independent of α. As $\|g_i\|_2$ is the *l*2 *norm* of the *i*th group, it can be further rewritten as $\|g_i\|_2 = \left(\alpha^T E^{(i)}\alpha\right)^{\frac{1}{2}}$, where $E^{(i)}$ is a diagonal indicator matrix with $E^{(i)}_{j,j}$ being 1 if $\alpha_j \in g_i$ and 0 otherwise. Hence Θ can be further rewritten as follows.

$$\Theta = \min_{\alpha} \quad \alpha^T\mathcal{X}_1^T\mathcal{X}_1\alpha - 2\mathbf{y}_1^T\mathcal{X}_1\alpha + \lambda\left(\|\alpha\|_1 + \sum_{i=1}^{n}\left(\alpha^T E^{(i)}\alpha\right)^{\frac{1}{2}}\right) + \mathbb{C}_1$$

Then the gradient *w.r.t* α can be derived as the following formula.

$$\frac{\partial\Theta}{\partial\alpha} = 2\mathcal{X}_1^T\mathcal{X}_1\alpha - 2\mathcal{X}_1^T\mathbf{y}_1 + \lambda\left(I(\alpha) + \sum_{i=1}^{n}\frac{E^{(i)}\alpha}{\left(\alpha^T E^{(i)}\alpha\right)^{\frac{1}{2}}}\right)$$

where $I(\alpha)$ is an indicator function for all entries of α, defined as $I(\alpha)_i = \frac{a_i}{|a_i|}$ and assigned as some particular value when $|\, a_i \,| = 0$ since $\|\alpha\|_1$ is not differentiable at zero entries [36].

Then the second derivative *w.r.t* α (*i.e.* the *Hessian matrix* H_α) can be further derived as follows.

$$H_\alpha = \frac{\partial^2\Theta}{\partial\alpha^2} = 2\mathcal{X}_1^T\mathcal{X}_1 + \lambda\Lambda$$

where Λ is the second derivative *w.r.t* the group lasso in the objective function Θ, *i.e.* $\sum_{i=1}^{n}\left(\alpha^T E^{(i)}\alpha\right)^{\frac{1}{2}}$. As $E^{(i)} = E^{(i)^T}$, it can be derived that

$$\begin{aligned}\Lambda \quad & = \sum_{i=1}^{n}\frac{E^{(i)}\left(\alpha^T E^{(i)}\alpha\right) - \left(E^{(i)}\alpha\right)\left(E^{(i)}\alpha\right)^T}{\left(\alpha^T E^{(i)}\alpha\right)^{\frac{3}{2}}} \\ & = \sum_{i=1}^{n}\frac{E^{(i)}\left(\alpha^T E^{(i)}\alpha\right) - \left(E^{(i)}\alpha\right)\left(E^{(i)}\alpha\right)^T}{\left(\|g_i\|_2\right)^3}\end{aligned}$$

Therefore, given any $x \in R^d$ with d being the dimensionality of α, it can be derived that

$$\begin{aligned}x^T H_\alpha x \quad & = 2x^T\mathcal{X}_1^T\mathcal{X}_1 x + \lambda x^T\Lambda x \\ & = 2(\mathcal{X}_1 x)^T(\mathcal{X}_1 x) + \lambda\sum_{i=1}^{n}\frac{x^T E^{(i)}\left(\alpha^T E^{(i)}\alpha\right)x - x^T\left(E^{(i)}\alpha\right)\left(E^{(i)}\alpha\right)^T x}{\left(\|g_i\|_2\right)^3} \\ & = 2(\mathcal{X}_1 x)^T(\mathcal{X}_1 x) + \lambda\sum_{i=1}^{n}\frac{\left(\alpha^T E^{(i)}\alpha\right)\left(x^T E^{(i)}x\right) - \left(\alpha^T E^{(i)}x\right)^T\left(\alpha^T E^{(i)}x\right)}{\left(\|g_i\|_2\right)^3}\end{aligned}$$

Since $\mathcal{X}_1 x$ is a column vector, it is evident that $(\mathcal{X}_1 x)^T(\mathcal{X}_1 x) = \|\mathcal{X}_1 x\|_2^2 \geqslant 0$. Additionally, as $E^{(i)}$ is a diagonal matrix with entries in $\{0, 1\}$ and $E^{(i)} = E^{(i)}E^{(i)} = E^{(i)^T}E^{(i)}$, $\left(\alpha^T E^{(i)}\alpha\right)\left(x^T E^{(i)}x\right) - \left(\alpha^T E^{(i)}x\right)^T\left(\alpha^T E^{(i)}x\right)$ can be rewritten as $\left(\left(E^{(i)}\alpha\right)^T\left(E^{(i)}\alpha\right)\right)\left(\left(E^{(i)}x\right)^T\left(E^{(i)}x\right)\right) - \left(\left(E^{(i)}\alpha\right)^T\left(E^{(i)}x\right)\right)^2$, with $E^{(i)}\alpha, E^{(i)}x$ both being column vectors. Then according to the *Cauchy–Schwarz inequality*, $\left(\alpha^T E^{(i)}\alpha\right)\left(x^T E^{(i)}x\right) - \left(\alpha^T E^{(i)}x\right)^T\left(\alpha^T E^{(i)}x\right)$ will keep non-negative and thus $\lambda\sum_{i=1}^{n}\frac{\left(\alpha^T E^{(i)}\alpha\right)\left(x^T E^{(i)}x\right) - \left(\alpha^T E^{(i)}x\right)^T\left(\alpha^T E^{(i)}x\right)}{\left(\|g_i\|_2\right)^3} \geqslant 0$, since $\|g_i\|_2$ and λ are always non-negative. And thus $x^T H_\alpha x$ will keep non-negative for any $x \in R^d$, meaning that H_α is a positive semi-definite *Hessian matrix*. Therefore, the image-view objective function Θ is convex, with its second-order necessary and sufficient conditions for convexity being guaranteed.

A.2. Convexity of the tag-view objective function

As mentioned previously in this paper, the following objective function of the tag-view linear sparse reconstruction is convex, meaning that there exists a global optimal β.

$$\Psi = \min_{\beta} \quad \|\mathbf{W}'\left(\mathbf{r} - \widehat{\mathbf{R}}\beta\right)\|_2^2 + \xi\|\beta\|_1$$

Similar to the former proof *w.r.t* the image-view objective function, we rewrite Ψ as follows to be a standard form of *Least Square Loss Problem* with a *l*1-norm regularizer.

$$\Psi = \min_{\beta} \quad \|\mathcal{X}_2\beta - \mathbf{y}_2\|_2^2 + \xi\|\beta\|_1$$

where

$$\mathcal{X}_2 = \left[\mathbf{W}'\widehat{\mathbf{R}}\right], \quad \mathbf{y}_2 = [\mathbf{W}'\mathbf{r}]$$

and then we can derive that

$$\begin{aligned}\Psi \quad & = (\mathcal{X}_2\beta - \mathbf{y}_2)^T(\mathcal{X}_2\beta - \mathbf{y}_2) + \xi\|\beta\|_1 \\ & = \beta^T\mathcal{X}_2^T\mathcal{X}_2\beta - 2\mathbf{y}_2^T\mathcal{X}_2\beta + \xi\|\beta\|_1 + \mathbb{C}_2\end{aligned}$$

where $\mathbb{C}_2$ is a constant independent of β. And the gradient *w.r.t* β can be derived as follows.

$$\frac{\partial\Psi}{\partial\beta} = 2\mathcal{X}_2^T\mathcal{X}_2\beta - 2\mathcal{X}_2^T\mathbf{y}_2 + \xi I(\beta)$$

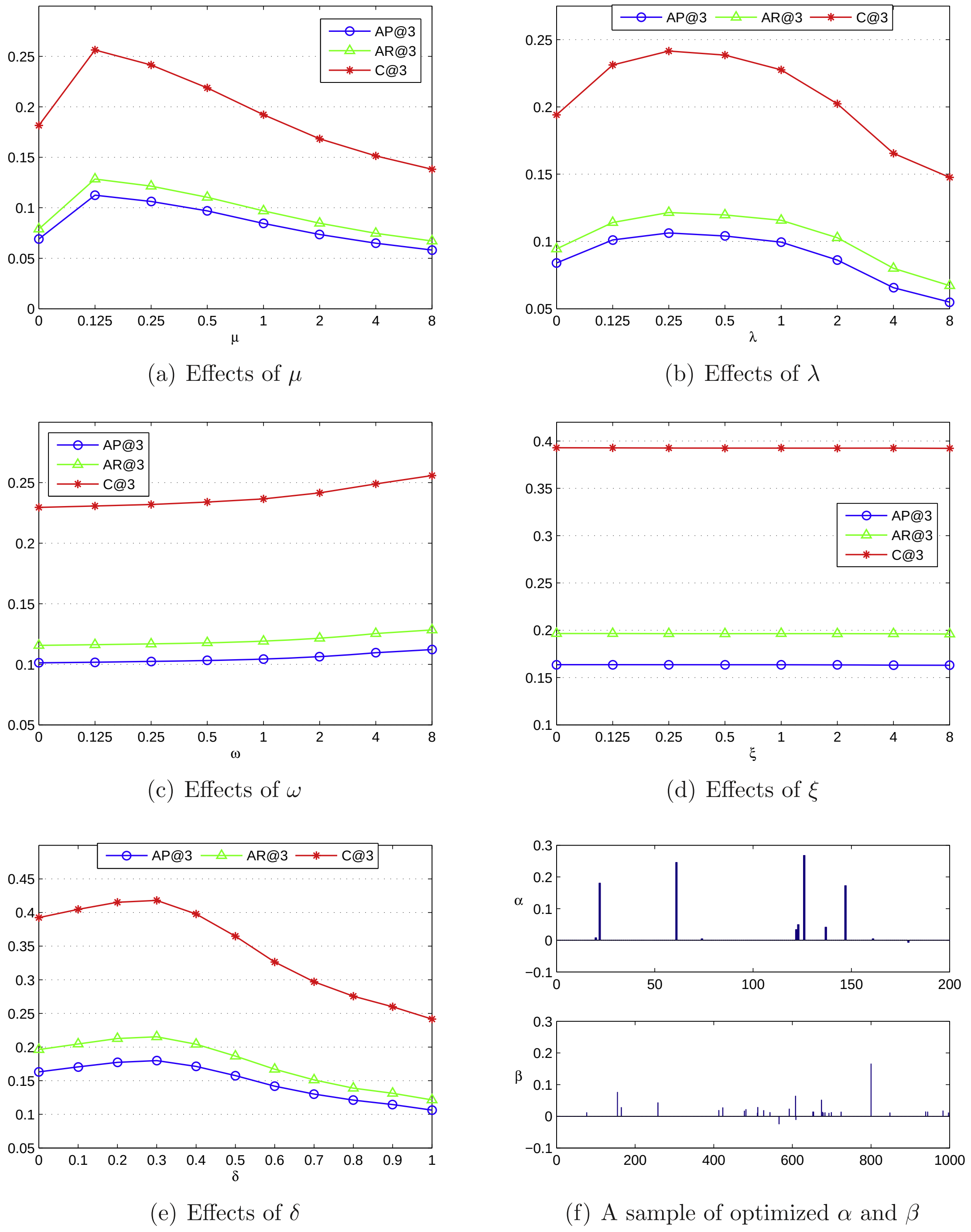
where $I(\beta)$ is an indicator function for all entries of β. Then the second derivative of β (*i.e.* the *Hessian matrix* H_β) can be further derived as follows.

$$H_\beta = \frac{\partial^2\Psi}{\partial\beta^2} = 2\mathcal{X}_2^T\mathcal{X}_2$$

For any $x \in R^{d_1}$ with d_1 being the dimensionality of $\beta, x^T H_\beta x = 2(\mathcal{X}_2 x)^T(\mathcal{X}_2 x) = 2\|\mathcal{X}_2 x\|_2^2 \geqslant 0$, meaning that H_β is a positive semi-definite *Hessian matrix*. Therefore, the tag-view objective function Ψ is convex, with its second-order necessary and sufficient conditions for convexity being guaranteed.

Appendix B. Parameter analyses for IAPR TC12, NUS-WIDE and Flickr30Concepts

The experimental results of parameter analyses for IAPR TC12, NUS-WIDE and Flickr30Concepts are respectively presented in Fig. B.8, B.9 and B.10. From the experimental results we can draw nearly the same conclusions as those on the benchmark Corel5k. Moreover, we find that the optimal parameter settings for $\mu, \lambda, \omega, \xi$ and δ on IAPR TC12 and Flickr30Concepts are respectively close to the ones that we select on the validate set of Corel5k, meaning that selected parameters on Corel5k can be well generalized to these two datasets with the same kinds of image features. And thus parameter settings of DLSR seem to be less dependent on datasets. Moreover, we find that the optimal parameter settings on NUS-WIDE, which provides different image features, are also mostly close to the ones that we select on the corresponding validate set.



(a) Effects of μ

(b) Effects of λ

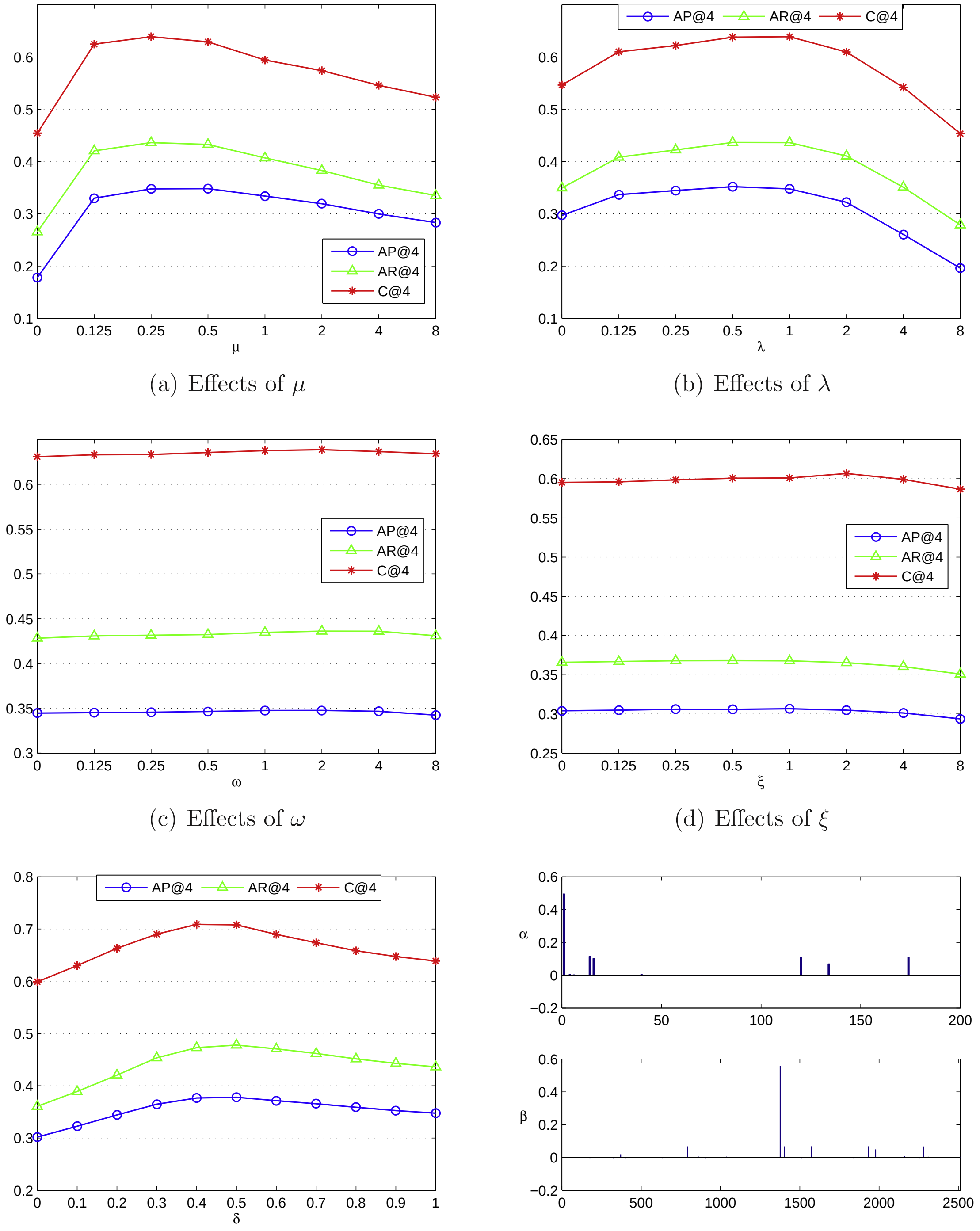
(c) Effects of ω

(d) Effects of ξ

(e) Effects of δ

(f) A sample of optimized α and β

Fig. B.9. Effects of μ, λ, ω in the objective function of image-view linear sparse reconstruction (sub-Fig. B.9(a)–(c)), ξ in the objective function of tag-view linear sparse reconstruction (sub-Fig. B.9(d)), and δ for combining the image-view and the tag-view reconstructed tagging vectors (sub-Fig. B.9(e)), in terms of $AP@3$, $AR@3$ and $C@3$ on the test set of the real-world NUS-WIDE, with sub-Fig. B.9(f) giving a sample of optimized α (upper) and β (lower).



(a) Effects of μ

(b) Effects of λ

(c) Effects of ω

(d) Effects of ξ

(e) Effects of δ

(f) A sample of optimized α and β

Fig. B.10. Effects of μ, λ, ω in the objective function of image-view linear sparse reconstruction (sub-Fig. B.10(a)–(c)), ξ in the objective function of tag-view linear sparse reconstruction (sub-Fig. B.10(d)), and δ for combining the image-view and the tag-view reconstructed tagging vectors (sub-Fig. B.10(e)), in terms of *AP*@4, *AR*@4 and *C*@4 on the test set of the real-world Flickr30Concepts, with sub-Fig. B.10(f) giving a sample of optimized α (upper) and β (lower).

References

[1] M. Ames, M. Naaman, Why we tag: motivations for annotation in mobile and online media, in: SIGCHI '07, 2007.

[2] B. Sigurbjörnsson, R.v. Zwol, Flickr tag recommendation based on collective knowledge, in: WWW '08, 2008.

[3] Z. Lin, G. Ding, M. Hu, J. Wang, X. Ye, Image tag completion via image-specific and tag-specific linear sparse reconstructions, in: CVPR '13, 2013.

[4] S. Feng, R. Manmatha, V. Lavrenko, Multiple bernoulli relevance models for image and video annotation, in: CVPR '04, 2004.

[5] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: ECCV '08, 2008.

[6] J. Liu, M. Li, Q. Liu, H. Lu, S. Ma, Image annotation via graph learning, Pattern Recognit. 42 (2) (2009).

[7] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: discriminative metric learning in nearest neighbor models for image auto-annotation, in: ICCV '09, 2009.

[8] Z. Ma, Y. Yang, F. Nie, J. Uijlings, N. Sebe, Exploiting the entire feature space with sparsity for automatic image annotation, in: MM '11, 2011.

[9] Z. Ma, F. Nie, Y. Yang, J.R.R. Uijlings, N. Sebe, Web image annotation via subspace-sparsity collaborated feature selection, IEEE Trans. Multimedia 14 (4) (2012) 1021–1030.

[10] A. Binder, W. Samek, K.-R. Mller, M. Kawanabe, Enhanced representation and multi-task learning for image annotation, Comput. Vis. Image Underst. 117 (5) (2013) 466–478.

[11] N. Garg, I. Weber, Personalized, interactive tag recommendation for flickr, in: RecSys '08, 2008.

[12] L. Wu, L. Yang, N. Yu, X. Hua, Learning to tag, in: WWW '09, 2009.

[13] S. Lee, W.D. Neve, K.N. Plataniotis, Y.M. Ro, Map-based image tag recommendation using a visual folksonomy, Pattern Recognit. Lett. 31 (9) (2010).

[14] A. Sun, S.S. Bhowmick, J.-A. Chong, Social image tag recommendation by concept matching, in: MM '11, 2011.

[15] W. Eom, S. Lee, W. De Neve, Y.M. Ro, Improving image tag recommendation using favorite image context, in: ICIP '11, 2011.

[16] Z. Qi, M. Yang, Z. Zhang, Z. Zhang, Mining partially annotated images, in: SIGKDD '11, 2011.

[17] Y. Jin, L. Khan, L. Wang, M. Awad, Image annotations by combining multiple evidence & wordnet, in: MM '05, 2005.

[18] C. Wang, F. Jing, L. Zhang, H. Zhang, Content-based image annotation refinement, in: CVPR '07, 2007.

[19] H. Xu, J. Wang, X. Hua, S. Li, Tag refinement by regularized lda, in: MM '09, 2009.

[20] S. Lee, W.D. Neve, Y.M. Ro, Tag refinement in an image folksonomy using visual similarity and tag co-occurrence statistics, Signal Process Image Commun. 25 (10) (2010) 761–773.

[21] S. Lee, W. Neve, Y. Ro, Image tag refinement along the 'what' dimension using tag categorization and neighbor voting, in: ICME '10, 2010.

[22] D. Liu, X. Hua, M. Wang, H. Zhang, Image retagging, in: MM '10, 2010.

[23] G. Zhu, S. Yan, Y. Ma, Image tag refinement towards low-rank, content-tag prior and error sparsity, in: MM '10, 2010.

[24] Y. Liu, F. Wu, Y. Zhang, J. Shao, Y. Zhuang, Tag clustering and refinement on semantic unity graph, in: ICDM '11, 2011.

[25] D. Liu, S. Yan, X. Hua, H. Zhang, Image retagging using collaborative tag propagation, IEEE Trans. Multimedia 13 (4) (2011) 702–712.

[26] G. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.

[27] L. Wu, R. Jin, A.K. Jain, Tag completion for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 716–727.

[28] X. Liu, S. Yan, T. Chua, H. Jin, Image label completion by pursuing contextual decomposability, ACM Trans. Multimed. Comput. Commun. Appl. 8 (2) (2012).

[29] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, D. Metaxas, Automatic image annotation using group sparsity, in: CVPR '10, 2010.

[30] C.C. Vogt, G.W. Cottrell, Fusion via a linear combination of scores, Inf. Retr. 1 (3) (1999) 151–173.

[31] J.A. Aslam, M. Montague, Bayes optimal metasearch: a probabilistic model for combining the results of multiple retrieval systems, in: SIGIR '00, 2000.

[32] M. Montague, J.A. Aslam, Relevance score normalization for metasearch, in: CIKM '01, 2001.

[33] J.A. Aslam, M. Montague, Models for metasearch, in: SIGIR '01, 2001.

[34] M.E. Renda, U. Straccia, Web metasearch: rank vs. score based rank aggregation methods, in: SAC '03, 2003.

[35] J. Liu, S. Ji, J. Ye, SLEP: Sparse Learning with Efficient Projections, Arizona State University, 2009. <http://www.public.asu.edu/jye02/Software/SLEP>.

[36] M. Schmidt, G. Fung, R. Rosales, Optimization methods for l1-regularization, University of British Columbia, Technical Report TR-2009-19, 2009.

[37] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, Nus-wide: a real-world web image database from national university of singapore, in: CIVR '09, 2009.

[38] M. Lux, S. Chatzichristofis, Lire: lucene image retrieval: an extensible java cbir library, in: MM '08, 2008.

[39] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: FOCS '06, 2006.

[40] Y. Gong, S. Lazebnik, Iterative quantization: a procrustean approach to learning binary codes, in: CVPR '11, 2011.

[41] K. He, F. Wen, J. Sun, K-means hashing: an affinity-preserving quantization method for learning binary compact codes, in: CVPR '13, 2013.

[42] Y. Lin, R. Jin, D. Cai, S. Yan, X. Li, Compressed hashing, in: CVPR '13, 2013.

[43] D. Liu, X.-S. Hua, L. Yang, M. Wang, H.-J. Zhang, Tag ranking, in: WWW '09, 2009.