Neurocomputing 138 (2014) 167–179

Contents lists available at ScienceDirect

Neurocomputing

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

Graph-based multimodal semi-supervised image classification

Wenxuan Xie, Zhiwu Lu, Yuxin Peng*, Jianguo Xiao

Institute of Computer Science and Technology, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 5 July 2013
Received in revised form
16 October 2013
Accepted 3 December 2013
Communicated by Haowei Liu
Available online 5 April 2014

Keywords:
Tag refinement
Graph-based label propagation
Support vector regression
Multiple graphs

ABSTRACT

We investigate an image classification task where training images come along with tags, but only a subset being labeled, and the goal is to predict the class label of test images without tags. This task is important for image search engine on photo sharing websites. In previous studies, it is handled by first training a multiple kernel learning classifier using both image content and tags to score unlabeled training images and then establishing a least-squares regression (LSR) model on visual features to predict the label of test images. Nevertheless, there remain three important issues in the task: (1) image tags on photo sharing websites tend to be imperfect, and thus it is beneficial to refine them for final image classification; (2) since supervised learning with a subset of labeled samples may be unreliable in practice, we adopt a graph-based label propagation approach by extra consideration of unlabeled data, and also an approach to combining multiple graphs is proposed; (3) kernel method is a powerful tool in the literature, but LSR simply treats the visual kernel matrix as an image feature matrix and does not consider the powerful kernel method. By considering these three issues holistically, we propose a graph-based multimodal semi-supervised image classification (GraMSIC) framework to handle the aforementioned task. Extensive experiments conducted on three publicly available datasets show the superior performance of the proposed framework.

1. Introduction

Image classification has been studied for decades [1–6]. The goal of image classification is to determine whether an image belongs to a predefined category or not. In the literature, different types of categories have been investigated, e.g., scenes [7] or objects [8]. To handle an image classification problem, a supervised framework can be used, where a binary classifier is first learned from manually labeled training images and then used to predict the class label of test images. By increasing the quantity and diversity of manually labeled images, the learned classifier can be enhanced. However, it is a time-consuming task to label images manually. Although it is possible to label large numbers of images for many categories for research purposes [9], it is usually unrealistic, e.g., in photo sharing applications. In practice, we usually have to handle a challenging classification problem by using only a small number of labeled samples. In the literature, semi-supervised learning [10] has been proposed to exploit the large number of unlabeled samples and thus helps to handle the scarcity of labeled samples to some extent.

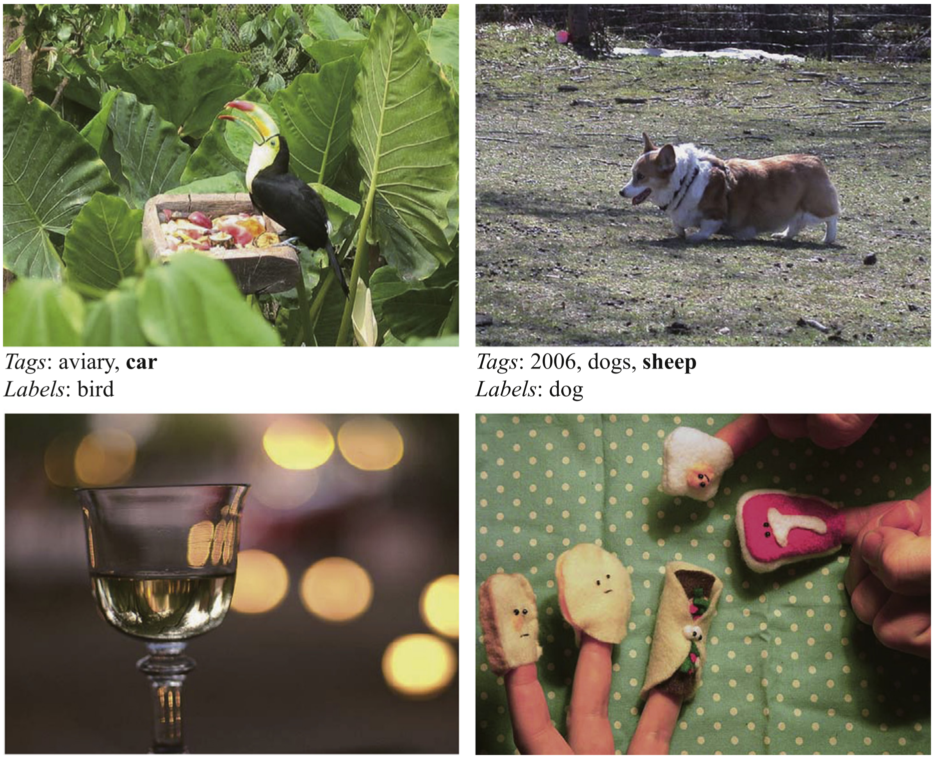
In this paper, we investigate a multimodal semi-supervised image classification problem originally raised in [11]. In this problem, training images have associated tags (e.g., from Flickr), and only a limited number of the training samples come along with class labels. The goal of this problem is to predict the class label of test images without tags. This is an important problem for image search engine on photo sharing websites. Since a newly uploaded image and also a considerable part of the existing images on websites have no associated tags, it is necessary to build up an image-only classifier for such image search engines with available resources (i.e., tagged images, and only a subset is labeled). To solve this problem, a two-step method has been proposed in [11]. In the first step, a multiple kernel learning (MKL) [12,13] classifier is learned by utilizing labeled training images with tags, which is then used to score unlabeled training images. In the second step, a least-squares regression (LSR) model is learned on the training set by using centered visual kernel columns as independent variables and using centered classification scores as dependent variables, which is then used to predict the scores of test images.

Nevertheless, we still need to consider the following *three* important issues, since they all may lead to performance degeneration in the aforementioned problem:

Tag imperfectness: Image tags on photo sharing websites (e.g., Flickr) are often inaccurate and incomplete, i.e., they may not directly relate to the image content and typically some relevant tags are missing. Some example images are shown in Fig. 1. For example, as we can see from the image on the upper left corner, the tag ‘car’ is inaccurate and the tag ‘bird’ is missing. Since the

* Corresponding author. Tel.: +86 10 82529699; fax: +86 10 82529207.
E-mail address: pengyuxin@pku.edu.cn (Y. Peng).

http://dx.doi.org/10.1016/j.neucom.2013.12.052
0925-2312/© 2014 Elsevier B.V. All rights reserved.



Tags: aviary, **car**
Labels: bird

Tags: 2006, dogs, **sheep**
Labels: dog

Tags: **tree**, reflection, bokeh, home
Labels: sunset, water

Tags: **food**
Labels: indoor, people

Fig. 1. Example images from PASCAL VOC'07 (top row) and MIR Flickr (bottom row) datasets with their associated tags and class labels. Tags in **bold** are inaccurate ones.

original tags are imperfect, it is a suboptimal choice to use them directly. Hence, we propose to refine these tags by using the affinity of image content as the first step.

Label scarcity: Since only a subset of the training images is labeled, supervised models such as an MKL classifier learned by using only labeled samples may be unreliable in practice. To handle the scarcity of labeled samples, we adopt a graph-based label propagation method to leverage the large number of unlabeled samples. By exploiting the graph structure of labeled and unlabeled samples, the label propagation method is shown to perform better in the experiments. More notably, since an average combination of multiple graphs for label propagation is only a suboptimal choice, we propose an approach to learning the combination weights of multiple graphs.

Ignorance of kernel method: The LSR model used in [11] simply treats the visual kernel matrix as an image feature matrix and does not consider the powerful kernel method. Moreover, the singular value decomposition (SVD) step involved in the LSR model is time-consuming. Instead of LSR, we propose to use support vector regression (SVR) to predict the class label of test images, since SVR can readily leverage the original visual kernel and make full use of image features in the reproducing kernel Hilbert space (RKHS) [14].

In summary, taking into account the *three* important issues, we propose a graph-based multimodal semi-supervised image classification (GraMSIC) framework to handle the aforementioned task by combining the following three components: (1) tag refinement; (2) graph-based label propagation by combining multiple graphs; (3) SVR. Fig. 2 shows the schematic overview of the proposed framework.

Upon our short conference version [15], this paper provides two additional contributions: (1) an approach to learning the combination weights of multiple graphs is proposed; (2) more extensive experimental results are added on three publicly available datasets, i.e., PASCAL VOC'07 [8], MIR Flickr [16] and NUS-WIDE-Object [17]. In the next two subsections, we briefly present preliminary notations and paper organization.

1.1. Preliminary notations

We denote training image set and test image set by $I_{tr} = \{x_1, x_2, \ldots, x_{n_1}\}$ and $I_{te} = \{x_{n_1+1}, x_{n_1+2}, \ldots, x_{n_1+n_2}\}$, respectively. Note that $n = n_1 + n_2$ is the total number of samples. Training images come along with tags, where the tag set is represented by $V = \{v_1, v_2, \ldots, v_m\}$ and m stands for the size of the tag set. The initial tag membership for all training images can be denoted by a binary matrix $T_{tr} \in \{0,1\}^{n_1 \times m}$ whose element $T_{tr}(i,j)$ indicates the presence of tag v_j in image x_i, i.e., $T_{tr}(i,j) = 1$ if tag v_j is associated with image x_i, and $T_{tr}(i,j) = 0$ otherwise. Moreover, only a small number of the training images are assigned with class labels from c categories, and the initial label matrix is denoted by $Y_{tr} \in \{1, 0, -1\}^{n_1 \times c}$, whose element $Y_{tr}(i,j)$ indicates the label of image x_i, i.e., $Y_{tr}(i,j) = 1$ if x_i is labeled as a positive sample of category j, $Y_{tr}(i,j) = -1$ if x_i is labeled negative, and $Y_{tr}(i,j) = 0$ if x_i is unlabeled. The goal is to predict the class label of test images without tags, i.e., an $n_2 \times c$ matrix Y_{te}.

Moreover, in order to state conveniently, the values determined by the learning algorithm are called 'parameters', and the values which require hand-tuning in advance are called 'hyperparameters' [18].

1.2. Paper organization

The paper is organized as follows. We begin by introducing related studies in the literature in Section 2. Then, we present the GraMSIC framework in Section 3. In Section 4, we discuss in detail the proposed approach to combining multiple graphs for label propagation. Moreover, we investigate the complexity issues and

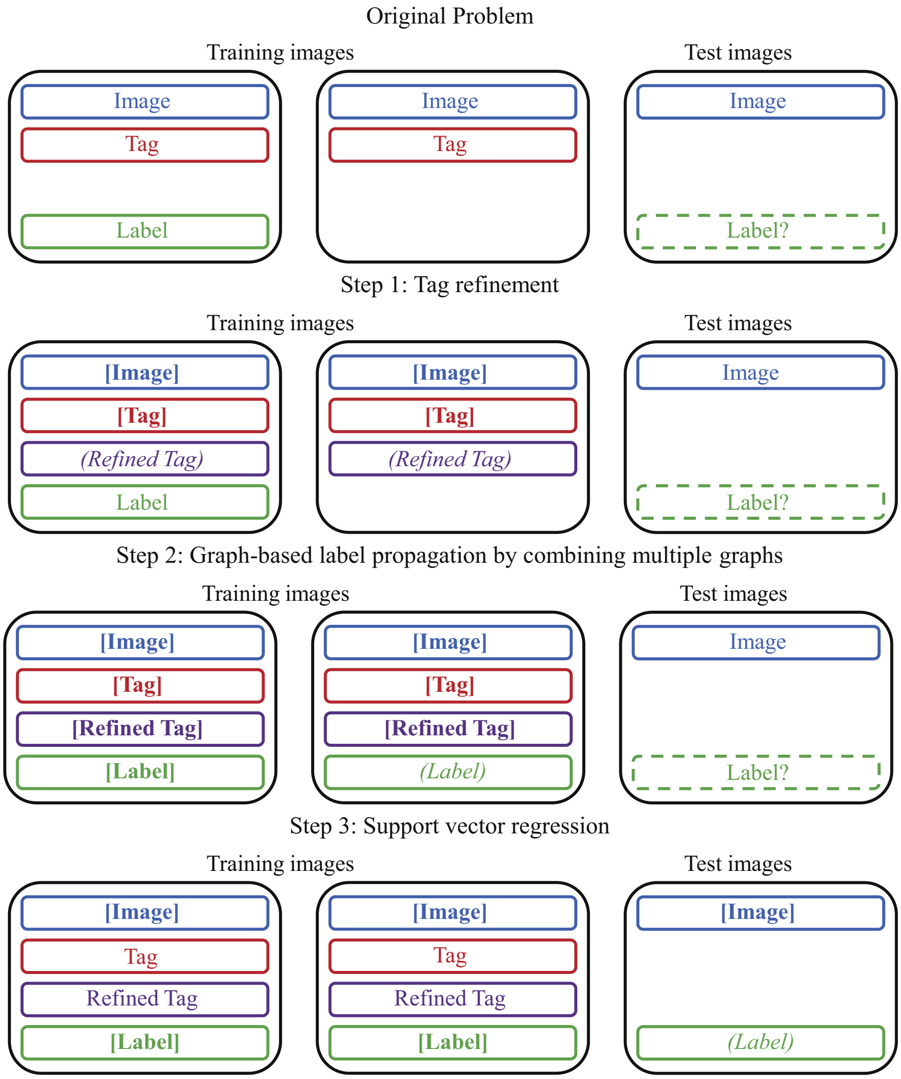


Fig. 2. Illustration of the proposed GraMSIC framework. Inputs and outputs of the corresponding step are denoted by **bold** words in square brackets and *italic* words in parentheses, respectively.

summarize our algorithm in Section 5. To evaluate the proposed framework, we report experimental results on three publicly available datasets in Section 6. Finally, Section 7 draws the conclusions.

2. Related work

2.1. Multimodal semi-supervised image classification

To the best of our knowledge, [11] has been the first attempt to handle the multimodal semi-supervised image classification task, where training images come along with tags, but only a subset being labeled, and the goal is to predict the class label of test images without tags. This task is illustrated in the uppermost subfigure of Fig. 2.

To handle this task, [11] proposes a two-step approach. In the first step, an MKL classifier is built up by using the labeled training images. The classifier is then used to predict the class label of other unlabeled training images with tags. Note that the original decision values instead of the classification results (i.e., 1 or -1) are used for the next step.

In the next step, a linear regression model is established by utilizing the visual kernel matrix and the decision values obtained in the previous step. In order to remove bias in the data, all the independent variables (i.e., visual kernel columns) and dependent variables (i.e., decision values of training samples) are normalized to zero mean. The regression model is then used to predict the scores of test images based on their visual features. A ranked list containing all the test images sorted by their predicted scores in descending order is returned as the final result.

2.2. Tag refinement

Since image tags on photo sharing websites (e.g., Flickr) tend to be inaccurate and incomplete, it is a necessary task to refine them. Based on the fundamental hypothesis that similar images should contain similar tags, neighbor voting [19] and kernel density estimation [20] approaches have been proposed. However, these two methods only take into account similar samples and do not consider dissimilar samples.

By considering both similar samples and dissimilar samples simultaneously, many more models have been proposed. Chen et al. [21] proposed to propagate tag probabilities based on a visual graph and a tag graph. Xu et al. [22] proposed a probabilistic graphical model named regularized latent Dirichlet allocation by exploiting both the statistics of tags and visual affinities. Moreover, a tag refinement approach based on low-rank and sparse matrix decomposition is proposed in [23]. Besides these, [24] proposes to refine tags by exploiting not only visual and tag similarity, but also the WordNet lexicon [25].

It should be noted that we adopt a simple and effective method by dealing with local and global consistency [26] for tag refinement, given that our objective in this paper is to tackle the *three* issues mentioned in Section 1 and to propose a more effective and efficient solution to the multimodal semi-supervised image classification problem.

2.3. *Graph-based learning*

Graph-based learning is a wide research area, and [10] is a comprehensive overview. Graph-based methods are often based on the hypothesis of label scarcity, where supervised methods may fail. Different from supervised models, graph-based methods pursue more robust results by leveraging the affinity of samples. Therefore, many graph-based learning methods are transductive. The Gaussian random fields and harmonic function method [27] and the local and global consistency method [26] are two well-known transductive approaches.

Besides the transductive methods, there are also inductive methods in the literature. Laplacian support vector machine (LapSVM) [28] is one of the inductive approaches. LapSVM incorporates a Laplacian regularizer into the support vector machine formulation, and the learned model can be directly used to predict a new test sample without retraining.

However, since our aim is to propose a novel graph-based solution to the multimodal semi-supervised image classification task, we only adopt the local and global consistency method [26] as the learning algorithm.

2.4. *Combining multiple graphs*

The affinity graph is an essential point of graph-based learning methods. In many recent real-world applications, there are multiple graphs of the same data. A key challenge under this setting is to combine different graphs automatically to achieve better predictive performance. In the literature, there are some studies which considered the task in different applications and demonstrated that integrating multiple graphs improve the prediction performance.

One of the first studies is in bioinformatics, where a label propagation approach to combining multiple graphs for protein function prediction is proposed [29]. This method is however not robust against noisy graphs, and a more robust probabilistic model named robust label propagation on multiple networks (RLPMN) is proposed [30]. Similarly, in genetic engineering, approaches to combining multiple graphs are proposed [31–33] by optimizing a predefined criterion named kernel-target alignment [34].

Besides, in the machine learning literature, an MKL-based algorithm is proposed to learn graph combination weights by integrating a graph regularizer into the formulation [35]. In the literature of multimedia content analysis, a method for integrating multiple graphs for the video annotation problem is proposed [36].

Recently, an algorithm has been proposed by taking into account sparse combination of multiple graphs [37]. As reported in [37], the proposed sparse multiple graph integration (SMGI) approach generally performs better than other existing methods. In cases where there are tens or hundreds of graphs, the approach in [37] will automatically select relevant graphs and ignore irrelevant graphs.

It should be noted that the current multimodal semi-supervised image classification task is different from the standard label propagation task. An approach which performs well in the standard label propagation task may not remain effective in the current task in this paper. In the current task, there are only three graphs in total, and thus sparse combination is relatively unsuitable. Actually, all the graph weights are nonzero in the experiments. Moreover, learning graph combination weights for each class separately is rather time-consuming. In order to accelerate the learning algorithm, we propose to learn graph combination weights for all the classes simultaneously.

3. The proposed framework

In this section, we present in detail the three components of the proposed GraMSIC framework in the following three subsections respectively, i.e., tag refinement in Section 3.1, graph-based label propagation in Section 3.2 and SVR in Section 3.3.

3.1. *Tag refinement*

As shown in Fig. 1, image tags on photo sharing websites (e.g., Flickr) tend to be inaccurate and incomplete, and thus directly using them may lead to inferior results. With this in mind, we propose to refine tags by using the affinity of image content as the first step. Although there have already been a series of studies on tag refinement in the literature [19–24], we adopt a simple and effective method by dealing with local and global consistency [26], given that our objective in this paper is to tackle the *three* issues mentioned in Section 1 and to propose a more effective and efficient solution to the multimodal semi-supervised image classification problem.

The local and global consistency method [26] propagates labels according to a graph. To handle the tag refinement problem, tags in the membership matrix T_{tr} are propagated by using a visual similarity graph. We denote the visual kernel of training samples by K_{tr}^v and adopt it as the graph. The normalized Laplacian of K_{tr}^v is defined as $L_{tr}^v = I - D^{-1/2} K_{tr}^v D^{-1/2}$, where D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th column of K_{tr}^v and I denotes an identity matrix. Therefore, the objective function for dealing with the tag refinement problem by using the local and global consistency method [26] is shown as

$$\min_{T_{tr^*}} (1-\alpha_1)\|T_{tr^*} - T_{tr}\|_F^2 + \alpha_1 \operatorname{tr}(T_{tr^*}^\top L_{tr}^v T_{tr^*}) \tag{1}$$

where α_1 is a regularization hyperparameter, and T_{tr^*} is the refined tag membership matrix. The first term of the above objective function is the Frobenius-norm constraint, and the second term is the Laplacian constraint, which means that a good refined tag representation should not change too much between similar images. By resorting to the analytical solution to Eq. (1) given by

$$T_{tr^*} = \left(I + \frac{\alpha_1}{1-\alpha_1} L_{tr}^v \right)^{-1} T_{tr} \tag{2}$$

we can obtain the refined tags T_{tr^*}.

3.2. *Graph-based label propagation*

After refining image tags, we have obtained a more precise similarity measure of training samples by learning to combine visual graph, tag graph and refined tag graph (which will be discussed at length later in Section 4). Here, we focus on the inference of the class label of unlabeled training images. As mentioned in Section 1, supervised models such as MKL may be unreliable by using only a limited number of labeled samples. Therefore, we adopt a graph-based label propagation method to tackle this problem by fully leveraging unlabeled samples. To be consistent with Section 3.1, we similarly adopt the local and global consistency method [26]. By denoting L as the combined graph Laplacian (which will be formally defined in Section 4), and thus we obtain the objective function for scoring unlabeled training images shown as

$$\min_{Y_{tr^*}}(1-\alpha_2)\| Y_{tr^*}-Y_{tr}\|_F^2+\alpha_2 \operatorname{tr}(Y_{tr^*}^\top L Y_{tr^*}) \tag{3}$$

where α_2 is also a regularization hyperparameter, and Y_{tr^*} denotes the predicted scores of all training samples. The closed-form solution of Eq. (3) is given by

$$Y_{tr^*}=\left(I+\frac{\alpha_2}{1-\alpha_2}L\right)^{-1}Y_{tr} \tag{4}$$

It should be noted that most of the elements in Y_{tr^*} have a small absolute value (i.e., close to 0), which may yield inferior final performance. To normalize the values in Y_{tr^*}, we use a simple algorithm shown in Eq. (5). Note that we define $Y_{tr^*}^1$ as the subset of Y_{tr^*} where the corresponding original labels in Y_{tr} equals 1 (i.e., positive), and we may similarly define $Y_{tr^*}^{-1}$ and $Y_{tr^*}^0$ as

$$\begin{aligned}
&Y_{tr^*}^1 \longleftarrow 1, \quad Y_{tr^*}^{-1}\longleftarrow -1\\
&Y_{tr^*}^0 \longleftarrow Y_{tr^*}^0-\tfrac{1}{2}(\max(Y_{tr^*}^0)+\min(Y_{tr^*}^0))\\
&Y_{tr^*}^0 \longleftarrow Y_{tr^*}^0/\max(Y_{tr^*}^0)
\end{aligned} \tag{5}$$

After the normalization step, the resultant Y_{tr^*} represents the predicted scores of all training samples.

3.3. Support vector regression

After obtaining scores of all training samples, the class label of test images can be inferred by resorting to a classification or regression model. Since the predicted scores of training samples are real-valued (i.e., the scores have not been quantized to 1 or -1), a regression model is preferred. In [11], SVD is performed on the centered kernel matrix for K_{tr}^v (i.e., each column of K_{tr}^v is normalized to 0 mean), and the regression coefficients can be computed by multiplying the pseudoinverse matrix of K_{tr}^v (which can be easily obtained after performing SVD) by the centered scores of training samples.

However, [11] simply treats each row of the visual kernel matrix as an individual image representation, and does not consider the powerful kernel method. Moreover, the SVD step is time-consuming. In order to directly leverage the kernel K_{tr}^v and to accelerate the learning algorithm, we propose to use SVR as the regression model. Similar to the SVM classifier, SVR can be kernelized to fully leverage image features in the RKHS along with the real-valued predicted scores of all training samples. The class label of test images predicted by SVR, i.e., Y_{te}, is the final result of the multimodal semi-supervised image classification problem addressed in this paper.

4. Learning to combine multiple graphs

In Section 3.2, we have already presented graph-based label propagation with multiple graphs. However, an average combination of multiple graphs for label propagation is only a suboptimal choice. Therefore, we will present in this section the approach to learning to combine multiple graphs. We first introduce the background and then discuss our approach in detail.

4.1. Background

After refining tags, we have obtained three graphs representing the training samples: the visual graph K_{tr}^v, the tag graph K_{tr}^t and the refined tag graph $K_{tr^*}^t$. Since we have these different data sources and they are likely to contain different information, we expect that effective integration of the complementary pieces of information will enhance the predictive performance. In order to combine multiple graphs, a natural choice is to take a weighted sum of the graph Laplacians [38]. By denoting L_{tr}^v, L_{tr}^t and $L_{tr^*}^t$ as the corresponding three Laplacians, and w_{tr}^v, w_{tr}^t and $w_{tr^*}^t$ as the combination weights, we can arrive at the following equations:

$$\begin{aligned}
&L=w_{tr}^v L_{tr}^v+w_{tr}^t L_{tr}^t+w_{tr^*}^t L_{tr^*}^t\\
&\mathbf{w}=[w_{tr}^v, w_{tr}^t, w_{tr^*}^t]^\top, \quad \mathbf{1}^\top\mathbf{w}=1, \ \ \mathbf{w}\ge 0
\end{aligned} \tag{6}$$

where L denotes the combined Laplacian, and $\mathbf{w}$ the vector of combination weights. To make things even clearer, we can further simplify Eq. (6) as

$$L=w_1L_1+w_2L_2+w_3L_3=\sum_{i=1}^{3} w_iL_i \tag{7}$$

where

$$\mathbf{w}=[w_1,w_2,w_3]^\top, \quad \mathbf{1}^\top\mathbf{w}=1, \ \ \mathbf{w}\ge 0 \tag{8}$$

Based on the aforementioned notations, the problem to be addressed can be formulated as follows:

$$\begin{aligned}
&\min_{Y_{tr^*},\mathbf{w}} \quad (1-\alpha_2)\|Y_{tr^*}-Y_{tr}\|_F^2+\alpha_2 \operatorname{tr}(Y_{tr^*}^\top L Y_{tr^*})\\
&\text{s.t.} \quad L=\sum_{i=1}^{3} w_iL_i, \quad \mathbf{1}^\top\mathbf{w}=1, \ \ \mathbf{w}\ge 0
\end{aligned} \tag{9}$$

Eq. (9) can be solved in a straightforward manner by iteratively optimizing Y_{tr^*} with $\mathbf{w}$ fixed and optimizing $\mathbf{w}$ with Y_{tr^*} fixed. However, the aforementioned formulation always leads to a degenerated result given by Eq. (10)

$$w_i=\begin{cases}1 & \operatorname{tr}(Y_{tr^*}L_iY_{tr^*})=\min(\operatorname{tr}(Y_{tr^*}L_jY_{tr^*})), \quad j=1,2,3\\ 0 & \text{otherwise,}\end{cases} \qquad i=1,2,3 \tag{10}$$

We can discover from Eq. (10) that the combined graph consequently degenerates to only one of the three graphs, which is an unsatisfactory result.

However, relatively fewer attempts have been made to tackle the graph combination problem in the literature. Tsuda et al. [29] proposed an algorithm which treats Lagrangian multipliers as combination weights. However, as declared by Kato et al. [30], the algorithm proposed in [29] tends to assign large weights to graphs which are less contributive to the classification task. In order to combine multiple graphs more robustly, the robust label propagation on multiple networks (RLPMN) [30] approach is proposed to tackle the following optimization problem:

$$\begin{aligned}
&\min_{\mathbf{f},\mathbf{w}} \quad \beta_y \sum_{i=1}^{l}(y_i-f_i)^2+\beta_{bias}\sum_{i=1}^{n} f_i^2+\beta_{net}\mathbf{f}L\mathbf{f}\\
&\text{s.t.} \quad L=\sum_{i=1}^{M} w_iL_i, \quad \mathbf{w}\ge 0
\end{aligned} \tag{11}$$

where M denotes the number of graphs, which equals 3 in our problem. It should be noted that the difference between Eq. (11) and our problem shown in Eq. (9) is threefold:

- Eq. (11) considers only one group of class labels (i.e., $\mathbf{f}$), while multiple groups of class labels (i.e., Y_{tr^*}) have been simultaneously taken into account in Eq. (9).
- We do not impose an L_2-norm regularizer on unlabeled data in Eq. (9).
- Combination weights $\mathbf{w}$ are normalized to sum to 1 in Eq. (9), whereas the resultant weights tend to be too large or too small in Eq. (11).

With these differences in mind and inspired by [30], we propose an approach to learning combination weights for the label propagation step in Eq. (9). Experimental results show that the approach to graph combination is beneficial for the later SVR step and the final performance of the multimodal semi-supervised image classification problem.

It should be noted that we have conducted extra experiments to combine different graphs. For convenience, we temporarily denote the visual graph by 'v', the original tag graph by 't', and the refined tag graph by 'r'. It can be observed that 'v+t+r' performs better than other combinations (i.e., 'v+t', 'v+r', and 't+r') in our experiments. A similar observation can also be found in another work dealing with supervised learning issues (see Table 3 in Reference [39]), where the combination of all three kinds of features yields the best performance. It may be due to the fact that the refined tags are derived by propagating visual affinity to the original tag representation, and thus 'r' is not a simple linear combination of 'v' and 't'. Therefore, the three graphs (i.e., 'v', 't' and 'r') are complementary to each other. With this in mind, adding 't' into 'v+r' can bring a further improvement due to such complementarity.

Our approach will be discussed in detail in the next subsection.

4.2. *Our approach to learning combination weights*

In this subsection, we formulate the graph combination problem in a probabilistic framework. We begin by establishing a probabilistic model for label propagation with a fixed Laplacian, and then introduce a prior of the weights. Finally, an EM algorithm is derived for maximum a posteriori (MAP) estimation according to the probabilistic model.

4.2.1. *Label propagation with a fixed Laplacian*

Here we give a probabilistic interpretation of label propagation with a fixed Laplacian. The label propagation method can be seen as an MAP estimation of the score matrix Y_{tr^*} in the probabilistic model described below. The score matrix Y_{tr^*} is in the set of model parameters. The observations Y_{tr} are drawn according to the Gaussian distribution

$$p(Y_{tr}(i,j)|Y_{tr^*}(:,j)) = \mathcal{N}\left(Y_{tr}(i,j); Y_{tr^*}(i,j), \frac{1}{1-\alpha_2}\right) \tag{12}$$

where $Y_{tr^*}(:,j)$ denotes the j-th column vector of Y_{tr^*}, and $\mathcal{N}(y;m,S)$ is a Gaussian probability density function of the observation y with mean m and covariance S defined as

$$\mathcal{N}(y;m,S) = \frac{1}{(2\pi)^{n/2}|S|^{1/2}} \exp\left(-\frac{1}{2}(y-m)^\top S^{-1}(y-m)\right) \tag{13}$$

where $|\cdot|$ denotes the determinant of a matrix. The prior of the model parameters is defined by the multivariate Gaussian distribution

$$p(Y_{tr^*}(:,j)) = \mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2}L^{-1}\right) \tag{14}$$

It should be noted that, since the Laplacian L is a positive semidefinite matrix but not a positive definite matrix, L^{-1} denotes the pseudoinverse matrix of L. MAP estimation pursues the value of the model parameters Y_{tr^*} which maximizes the posterior probability

$$\prod_{j=1}^{c} p(Y_{tr^*}(:,j)|Y_{tr}(:,j)) = \prod_{j=1}^{c} \frac{p(Y_{tr^*}(:,j))\prod_{i=1}^{n_1} p(Y_{tr}(i,j)|Y_{tr^*}(:,j))}{p(Y_{tr}(:,j))} \tag{15}$$

Since the denominator of Eq. (15) is constant for maximization, the MAP estimation is equivalent to maximizing the following objective function:

$$\sum_{j=1}^{c}\left(\log\ p(Y_{tr^*}(:,j)) + \sum_{i=1}^{n_1}\log\ p(Y_{tr}(i,j)|Y_{tr^*}(i,j))\right)$$

$$= -\frac{1}{2}\sum_{j=1}^{c}(\alpha_2 Y_{tr^*}(:,j)^\top L Y_{tr^*}(:,j) + (1-\alpha_2)\|Y_{tr^*}(:,j) - Y_{tr}(:,j)\|^2) + C$$

$$= -\frac{1}{2}(\alpha_2\ \mathrm{tr}(Y_{tr^*}^\top L Y_{tr^*}) + (1-\alpha_2)\|Y_{tr^*} - Y_{tr}\|_F^2) + C \tag{16}$$

where C denotes a constant value irrelevant to the score matrix Y_{tr^*}. The value of C is shown as follows:

$$C = -\frac{cn_1(n_1+1)}{2}\log(2\pi) - \frac{c}{2}\log\left|\alpha_2^{-1}L^{-1}\right| + \frac{cn_1}{2}\log(1-\alpha_2) \tag{17}$$

We can see from Eq. (16) that the values of Y_{tr^*} at the maximum of the posterior probability are equal to the solution of Eq. (9). The validity of the aforementioned equivalence is due to the proper selection of the prior distribution (Eq. (14)) and the likelihood function (Eq. (12)), both of which are key components of a Bayesian probabilistic model.

More notably, if we replace the Laplacian L in Eq. (14) by Eq. (7), we can arrive at the following equation:

$$p(Y_{tr^*}(:,j)) = \frac{1}{Z}\prod_{i=1}^{3}\mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2 w_i}L_i^{-1}\right) \tag{18}$$

where Z is a normalizing constant defined as follows:

$$Z = \frac{(2\pi)^{-n_1}|\alpha_2^{-1}L^{-1}|}{\prod_{i=1}^{3}|\alpha_2^{-1}w_i^{-1}L_i^{-1}|} \tag{19}$$

From Eq. (18), we can observe that the prior distribution of the model parameters for the fixed weight combination of multiple graphs is expressed as the product of multiple Gaussians. This formulation facilitates the development of the probabilistic model of the graph combination algorithm.

4.2.2. *Prior distribution over graph weights*

As described above, we have obtained the probabilistic model for label propagation with a fixed Laplacian. Here we investigate the situation where the graph weights are unknown. We introduce a prior of the graph weights and marginalize out the random variables of the weights from the expressions. To begin with, we employ the Gamma distribution for the prior of the weights. The Gamma distribution is defined as

$$\mathrm{Gamma}(w;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}w^{\alpha-1}\exp(-\beta w) \tag{20}$$

where $w \geq 0$, $\alpha \geq 0$, $\beta \geq 0$. In the probabilistic model described here, each component (i.e., a Gaussian distribution) of Eq. (18)

$$\mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2 w_i}L_i^{-1}\right) \tag{21}$$

is substituted by an infinite mixture of Gaussians

$$\int_0^\infty \mathrm{Gamma}\left(w_i; \frac{1}{2}\nu, \frac{1}{2}\nu\right)\mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2 w_i}L_i^{-1}\right)dw_i \tag{22}$$

where ν is a positive hyperparameter. In Eq. (22), the mixture coefficients are expressed by the Gamma distribution, and the weights $\mathbf{w} = [w_1, w_2, w_3]^\top$ can be seen as latent variables. The prior distribution of the graph weights is flatter if ν is chosen to be smaller.

4.2.3. *EM algorithm for MAP estimation*

With the Gamma distribution being the prior of the graph weights, we present an EM algorithm for MAP estimation of model parameters Y_{tr^*}. Given that the hyperparameters α_2 and ν are fixed in advance, MAP estimation finds the model parameters by maximizing Eq. (16):

$$\sum_{j=1}^{c}\left(\log\ p(Y_{tr^*}(:,j)) + \sum_{i=1}^{n_1}\log\ p(Y_{tr}(i,j)|Y_{tr^*}(i,j))\right)$$

By considering Eq. (22), the logarithm of the prior distribution is rewritten to be

$$\sum_{j=1}^{c}(\log\ p(Y_{tr^*}(:,j))) = \log\ Z + \sum_{j=1}^{c}\sum_{i=1}^{3}\log\int_0^{\infty} h_i(Y_{tr^*}(:,j), w_i)\ dw_i \tag{23}$$

where Z is a normalizing constant defined as

$$Z = cn_1\ \log(2\pi) - c\ \log|\alpha_2^{-1}L^{-1}| + c\sum_{i=1}^{3}\log|\alpha_2^{-1}w_i^{-1}L_i^{-1}| \tag{24}$$

and the function $h_i(\cdot,\cdot)$ is defined as

$$h_i(Y_{tr^*}(:,j), w_i) = \text{Gamma}\left(w_i; \frac{1}{2}\nu, \frac{1}{2}\nu\right)\mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2 w_i}L_i^{-1}\right) \tag{25}$$

It should be noted that the EM algorithm consists of E-step and M-step: E-step computes the optimal distribution denoted by $r(w_i)$, and M-step maximizes the logarithm of the posterior probability with respect to the model parameters Y_{tr^*}.

Based on the aforementioned definitions, we can derive the optimal distribution $r(w_i)$ to be computed in E-step by using variational analysis as follows:

$$\begin{aligned}\log\ r(w_i) &= \sum_{j=1}^{c}\log\ h_i(Y_{tr^*}(:,j), w_i) + C\\
&= \sum_{j=1}^{c}\log\ \text{Gamma}\left(w_i; \frac{1}{2}\nu, \frac{1}{2}\nu\right)\mathcal{N}\left(Y_{tr^*}(:,j); \mathbf{0}, \frac{1}{\alpha_2 w_i}L_i^{-1}\right) + C\\
&= -\frac{\alpha_2 w_i}{2}\sum_{j=1}^{c} Y_{tr^*}(:,j)^{\top} L_i Y_{tr^*}(:,j) + \frac{cn_1}{2}\log\ w_i\\
&\quad + \left(\frac{c\nu}{2} - c\right)\log\ w_i - \frac{c\nu}{2}w_i + C\\
&= \log\ \text{Gamma}\left(w_i; \frac{c\nu + cn_1}{2} - c + 1, \frac{c\nu}{2} + \frac{\alpha_2}{2}\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})\right)\end{aligned} \tag{26}$$

where C denotes the terms independent of w_i. Since the expectation of Gamma distribution $\text{Gamma}(w;\alpha,\beta)$ is $\overline{w} = \alpha/\beta$ [40], we can obtain the expectation of w_i over the optimal distribution $r(w_i)$ by

$$\overline{w}_i = \int_0^{\infty} w_i r(w_i)\ dw_i = \frac{c\nu + cn_1 - 2c + 2}{c\nu + \alpha_2\ \text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})} \tag{27}$$

It can be concluded from Eq. (9) that the importance of a graph is determined by the trace norm $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})$. With this in mind, a graph with a large $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})$ may dominate the final result. Therefore, large $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})$ should be penalized to allow for a better combination. Since the term $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})$ is in the denominator of Eq. (27), the weights of graphs with large $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})$ will be small. After obtaining the graph weights $\mathbf{w}$, we can compute Y_{tr^*} according to the following analytical solution:

$$Y_{tr^*} = \left(I + \frac{\alpha_2}{1-\alpha_2}L\right)^{-1} Y_{tr} \tag{28}$$

which can also be viewed as the M-step. However, there is still a problem to be addressed: the resultant graph weights $\mathbf{w}$ may be too large or too small if we simply iterate between Eqs. (27) and (28) until convergence. Fortunately, Eq. (9) is equivalent to the following objective function:

$$\begin{aligned}\min_{Y_{tr^*},\mathbf{w}}\quad & \lambda(1-\alpha_2)\|Y_{tr^*} - Y_{tr}\|_F^2 + \lambda\alpha_2\ \text{tr}(Y_{tr^*}^{\top}LY_{tr^*})\\
\text{s.t.}\quad & L = \sum_{i=1}^{3} w_i L_i, \quad \mathbf{1}^{\top}\mathbf{w} = 1, \quad \mathbf{w} \geq 0\end{aligned} \tag{29}$$

where λ is an arbitrary positive factor. Therefore, we can rewrite Eq. (27) as follows:

$$w_i = \frac{c\nu + cn_1 - 2c + 2}{c\nu + \lambda\alpha_2\ \text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*})} \tag{30}$$

Since the Laplacian L_i is a positive semidefinite matrix, the expression $\text{tr}(Y_{tr^*}^{\top}L_i Y_{tr^*}) \geq 0$ always holds. Therefore, w_i in Eq. (30) is a monotonic decreasing function with respect to λ, and thus we can search for the appropriate λ by using Newton's method in order to let the expression $\mathbf{1}^{\top}\mathbf{w} = 1$ hold true. Finally, the EM algorithm for learning the graph weights for label propagation is summarized as follows:

E-step: Update $\mathbf{w}$ using Eq. (30) by searching for the appropriate λ via Newton's method to let the expression $\mathbf{1}^{\top}\mathbf{w} = 1$ hold true.

M-step: Update Y_{tr^*} using Eq. (28).

The two steps are repeated until convergence. EM algorithms are guaranteed to converge to a local optimum [41], so is the aforementioned algorithm. Currently, we only select the equal weights as the initial point. It should be noted that a multipoint search strategy may be adopted to further improve the performance, although it will increase the computational complexity. More notably, instead of learning model parameters for c classes simultaneously, we can learn to combine multiple graphs for each class separately, although this increases the computational cost.

5. Complexity issues and algorithm summary

We begin by analyzing the complexity issues in this section. Recall that the sample size is denoted by n. Since training sample size and test sample size have the same orders of magnitude, we do not explicitly distinguish between them. The method proposed in [11] consists of an MKL classifier and an LSR model. Since the MKL classifier is built upon a limited number of samples (i.e., no more than 200 in our experiments), the computational cost of the training and inference steps is negligible. However, the SVD of the centered visual kernel matrix involved in the LSR is time-consuming, where the time complexity is $O(n^3)$.

As a comparison, the proposed GraMSIC framework is made up of three components: tag refinement, graph-based label propagation by combining multiple graphs and SVR. The most time-consuming step is the inversion of an $n \times n$ matrix when computing the analytical solution to a semi-supervised problem, where the time complexity is $O(n^3)$. However, we can adopt the iterative steps suggested in [26] to accelerate the semi-supervised learning, and thus the computational complexity of the label propagation algorithm can be reduced to $O(n^2)$ with respect to the data size n.

As for the third component (i.e., SVR), the complexity is also $O(n^2)$, since the LIBSVM implementation [42] we adopt is a decomposition-based algorithm [43]. As a consequence, the total computational complexity of the proposed GraMSIC framework is $O(n^2)$, whereas the method in [11] has a time complexity of $O(n^3)$. Therefore, the proposed GraMSIC framework can perform more efficiently.

Moreover, as a summarization of the above discussion, the proposed GraMSIC framework is shown in Algorithm 1.

Algorithm 1. The proposed GraMSIC framework.

```
Input:
    Visual kernel of training samples $K_{tr}^{v} \in \mathbb{R}^{n_1 \times n_1}$
    Visual kernel of test samples $K_{te}^{v} \in \mathbb{R}^{n_2 \times n_1}$ (each value in this
    matrix is computed using a training sample and a test
    sample)
    Tag membership matrix $T_{tr} \in \{0,1\}^{n_1 \times m}$
    Label matrix of training samples $Y_{tr} \in \{1,0,-1\}^{n_1 \times c}$
    Hyperparameters $\alpha_1$, $\alpha_2$, $\nu$, $C_{reg}$
Output:
    Label matrix of test samples $Y_{te} \in \{1,0,-1\}^{n_2 \times c}$
    Compute $L_{tr}^{\nu}$, the Laplacian of $K_{tr}^{\nu}$.
    Obtain refined tags $T_{tr^*}$ by solving Eq. (1).
    Initialize graph combination weights $\mathbf{w} = [1/3, 1/3, 1/3]$.
    repeat
        Compute Laplacian $L = \sum_{i=1}^{3} w_i L_i$.
        Compute predicted labels of training samples $Y_{tr^*}$ using
        Eq. (28).
        Compute $\mathbf{w}$ using Eq. (30) by searching for the
        appropriate $\lambda$ via Newton's method to let the expression
        $\mathbf{1}^{\top}\mathbf{w} = 1$ hold true.
    until convergence
    Normalize $Y_{tr^*}$ according to Eq. (5).
    Train an SVR model using $K_{tr}^{\nu}$ and normalized $Y_{tr^*}$.
    Predict $Y_{te}$ by using the trained SVR model along with $K_{te}^{\nu}$.
```

6. Experimental results

We conduct extensive experiments to evaluate the effectiveness of the proposed GraMSIC framework. In this section, we begin by describing the experimental setup and the evaluation metric. Secondly, we evaluate the effectiveness of each component of the proposed GraMSIC framework. Thirdly, we compare the proposed approach with the state-of-the-art graph combination algorithms [37,30]. Finally, we present the hyperparameter tuning details and discuss the complexity issues.

6.1. Experimental setup

The experiments are conducted on three publicly available datasets, i.e., the PASCAL VOC'07 [8], the MIR Flickr [16] and the NUS-WIDE-Object [17]. In particular, there are 9963 images with 804 tags from 20 categories in the PASCAL VOC'07 dataset, 25,000 images with 457 tags from 38 categories in the MIR Flickr dataset, and 30,000 images with 1000 tags from 31 categories in the NUS-WIDE-Object dataset. In addition, the PASCAL VOC'07 dataset is split into a training set of 5011 images and a test set of 4952 images, the MIR Flickr dataset is equally split into a training set of 12,500 images and a test set of 12,500 images, and the NUS-WIDE-Object dataset is split into a training set of 17,928 images and a test set of 12,072 images.

Note that both the PASCAL VOC'07 dataset and the MIR Flickr dataset have been used in [11]. There are $P=15$ different image representations and a tag membership matrix publicly available on these two datasets. The 15 different image representations are derived from two local descriptors (SIFT, Hue), three global color histograms (RGB, HSV and LAB) and a GIST descriptor. Fig. 3 illustrates all the aforementioned image representations. We use the same visual kernel as that in [11]. Specifically, we average the distances between images based on these different representations, and use it to compute an RBF kernel, which is shown as

$$k^{\nu}(x_i, x_j) = \exp(-\lambda^{-1} d(x_i, x_j)) \tag{31}$$

where the scale factor λ is set to the average pairwise distance, i.e., $\lambda = n^{-2}\sum_{i,j=1}^{n} d(x_i, x_j)$, and $d(x_i, x_j) = \sum_{p=1}^{P} \lambda_p^{-1} d_p(x_i, x_j)$, where the scale factor is defined as $\lambda_p = \max_{i,j} d_p(x_i, x_j)$. Following the settings in [11], we adopt L1 distance for the color histograms, L2 for GIST, and χ^2 for the visual word histograms. Moreover, we compute the cosine similarity kernel for tag features.

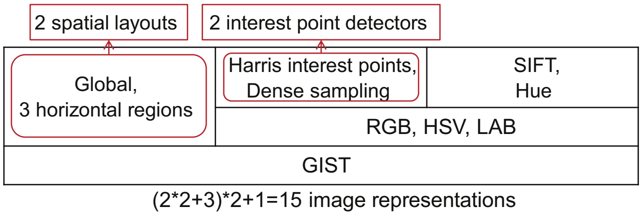


Fig. 3. Fifteen image representations [11] used in the PASCAL VOC'07 and the MIR Flickr datasets. SIFT [44] and Hue are extracted with two interest point detectors and two spatial layouts. RGB, HSV, and LAB are extracted with two spatial layouts. GIST [45] is extracted globally.

As for the NUS-WIDE-Object dataset, we adopt the 500-dimensional bag of words based on SIFT descriptions publicly available in the dataset and compute a χ^2 kernel. Moreover, to be in accordance with the aforementioned two datasets, we also compute the cosine similarity kernel for tag features for the NUS-WIDE-Object dataset.

There are four tunable hyperparameters in our model, i.e., α_1, α_2, ν and the regularization hyperparameter of SVR denoted by C_{reg}. The setting of these hyperparameters will be investigated in Section 6.5.

6.2. Evaluation metric

In our experiments, we evaluate results by using the mean average precision (mAP) over all classes. To be in accordance with [11], we adopt the evaluation criterion in the PASCAL VOC challenge evaluation [8], which is given as

$$AP = \frac{1}{11}\sum_{r} P(r) \tag{32}$$

where $P(r)$ denotes the maximum precision over all recalls larger than $r \in \{0, 0.1, 0.2, \ldots, 1.0\}$. A larger value indicates a better performance. It should be noted that all the AP scores are computed based on the ranked lists of all test samples.

6.3. Evaluation of the GraMSIC framework

Since the proposed GraMSIC framework consists of three components (i.e., tag refinement, graph-based label propagation by combining multiple graphs and SVR), we conduct experiments to demonstrate the effectiveness of each of the three components respectively. Concretely, we compare the following four approaches:

- MKL+LSR[11]: An MKL classifier learned on labeled training samples, followed by least-squares regression on the MKL scores for all training samples to obtain the visual classifier.
- GLP+SVR(ours): A graph-based label propagation approach based on a combined graph Laplacian L by *averagely* fusing visual graph and tag graph, followed by SVR on the normalized decision values of all training samples to predict the scores of test samples.
- TR+GLP+SVR(ours): Tag refinement by using the local and global consistency method [26], followed by a graph-based label propagation method based on a combined graph Laplacian L by *averagely* fusing visual graph, tag graph and refined tag graph. Finally, SVR is learned on the normalized decision values.
- TR+GLP*+SVR(ours): Tag refinement by using the local and global consistency method [26], followed by a graph-based label propagation method by combining multiple graphs which simultaneously learns predicted scores and graph weights. Finally, SVR is learned on the normalized decision values.

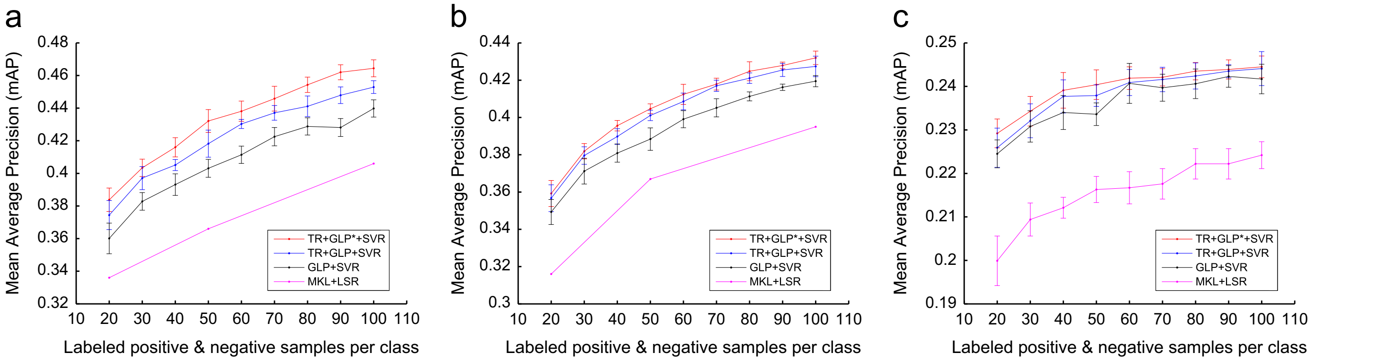


Fig. 4. Performance in mAP (mean ± standard deviation) on the three datasets using varied number of labeled examples for each class. (a) PASCAL VOC'07. (b) MIR Flickr. (c) NUS-WIDE-Object.

It should be noted that there is also a related paper [46] on multi-label image classification using the same datasets, where the authors assume that the class label vector (i.e., all class label assignments) is known for some given samples. Nevertheless, following the problem settings in [11], positive and negative samples are randomly chosen for only one class at a time. Most probably, different labeled samples are chosen for different classes, and thus the aforementioned problem does not belong to a multi-label classification problem. Due to different settings of input class labels, we do not make direct comparisons with the results in [46].

We randomly select n_l positive and the same number of negative samples for each class and all the rest are unlabeled. As n_l varies, the mAP scores for all the three datasets are illustrated in Fig. 4.

Since the sampling strategy of labeled training images may affect the final result, the averaged performance over 10 random samplings is reported in the figures. Besides that the performance improves as the number of labeled samples increases, we have the following three additional observations from Fig. 4:

- *GLP+SVR* performs much better than *MKL+LSR* on all datasets. It may be due to that, although MKL [13] is a powerful algorithm, it performs unsatisfactorily when using only a limited number of labeled samples. In contrast, graph-based label propagation is good at dealing with such problems. Moreover, the LSR model does not consider the powerful kernel method, while SVR can readily utilize the original visual kernel and thus leverage its full power.
- *TR+GLP+SVR* performs significantly better[1] than *GLP+SVR* on the PASCAL VOC'07 dataset and the MIR Flickr dataset, and slightly better on the NUS-WIDE-Object dataset. Since the original tags tend to be inaccurate and incomplete, directly using them may lead to inferior results. Therefore, refining the initial tags with the help of the visual content is beneficial for the final performance.
- Compared to *TR+GLP+SVR*, *TR+GLP*+SVR* performs significantly better on the PASCAL VOC'07 dataset, and slightly better on the MIR Flickr dataset and the NUS-WIDE-Object dataset. These results show that the proposed approach to graph combination performs more effectively than a simple average combination of multiple graphs for label propagation.

As a consequence, the GraMSIC framework (i.e., *TR+GLP*+SVR*) performs significantly better than the existing *MKL+LSR* approach [11], due to the effectiveness of the three components. It should be noted that only a slight improvement is observed on the NUS-WIDE-Object dataset after adding the tag refinement component and the graph combination component, which may be due to the fact that the NUS-WIDE-Object dataset (mAP is less than 0.25) is more challenging than the other two datasets (mAP is around 0.4). It is beneficial to take into account both tag refinement and graph combination on such a challenging dataset, but the improvement may be limited.

We also note that, since there are no published results of the *MKL+LSR* approach [11] on the NUS-WIDE-Object dataset, we implement the algorithm by using the MKL code [13] publicly available.[2] Moreover, to make a detailed comparison, we report the per-class results of the proposed GraMSIC framework along with the published results in [11] on the PASCAL VOC'07 dataset in Table 1 and the MIR Flickr dataset in Table 2 using 50 positive and 50 negative labeled examples for each class. We can observe from these two tables that (1) *TR+GLP*+SVR* (i.e., the proposed GraMSIC framework) outperforms other methods for most of the categories; (2) as for the other categories, the differences between the results of *TR+GLP*+SVR* and the best ones are relatively small.

6.4. Comparison with state-of-the-art graph combination algorithms

Recall that the problem we investigate in this paper is an image classification task where training images come along with tags, but only a subset being labeled, and the goal is to predict the class label of test images without tags. This task, as illustrated in the uppermost subfigure of Fig. 2, is different from many other image classification problems. Therefore, there are few related studies addressing this task in the literature.

In this subsection, we make comparisons between the graph combination approach in the GraMSIC framework and the state-of-the-arts [37,30]. We compare with the recent algorithm named sparse multiple graph integration (SMGI) [37], since it is reported to perform better than other methods [29–33,35,36] by taking into account the sparse constraints. Moreover, we compare with the robust label propagation on multiple networks (RLPMN) algorithm [30], since the proposed graph combination approach is inspired by [30] and is most related to [30]. It should be noted that we conduct all the experiments in the GraMSIC framework (i.e., *TR+GLP*+SVR*), and the only difference is the graph combination methods.

As a quantitative comparison, the mAP scores for all the three datasets are illustrated in Fig. 5 using varied number of labeled examples for each class. It can be observed from Fig. 5 that, in the multimodal semi-supervised image classification task, the proposed graph combination approach performs better than

[1] The significance is judged by the paired t-test with a significance level of 0.05.

[2] http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html.

Table 1
AP scores for all the classes using 50 positive and 50 negative labeled examples for each class on the PASCAL VOC'07 dataset.

Methods	Aeroplane	Bicycle	Bird	Boat	Bottle
MKL+LSR [11]	0.5920	0.3240	0.3760	0.5190	0.1540
GLP+SVR	0.6272 ± 0.0073	0.3998 ± 0.0251	0.4035 ± 0.0143	0.5509 ± 0.0226	0.1585 ± 0.0272
TR+GLP+SVR	0.6396 ± 0.0077	0.4216 ± 0.0256	0.4123 ± 0.0197	0.5579 ± 0.0187	0.1613 ± 0.0283
TR+GLP*+SVR	**0.6509 ± 0.0217**	**0.4597 ± 0.0247**	**0.4361 ± 0.0091**	**0.5869 ± 0.0107**	**0.1822 ± 0.0471**
Bus	Car	Cat	Chair	Cow	Diningtable
0.2780	0.5010	0.3660	**0.3000**	0.1170	0.2550
0.3765 ± 0.0287	0.5056 ± 0.0314	0.4020 ± 0.0204	0.2791 ± 0.0372	0.2443 ± 0.0199	**0.2964 ± 0.0263**
0.4104 ± 0.0256	0.5248 ± 0.0362	0.4274 ± 0.0186	0.2965 ± 0.0304	0.2636 ± 0.0238	0.2886 ± 0.0367
0.4299 ± 0.0252	**0.5686 ± 0.0187**	**0.4417 ± 0.0187**	0.2857 ± 0.0393	**0.2802 ± 0.0328**	0.2777 ± 0.0125
Dog	Horse	Motorbike	Person	Pottedplant	Sheep
0.3310	0.6370	0.3830	**0.7030**	0.2120	0.2180
0.3371 ± 0.0247	0.6677 ± 0.0137	0.4466 ± 0.0362	0.6664 ± 0.0234	0.2151 ± 0.0373	0.3220 ± 0.0178
0.3582 ± 0.0175	0.6885 ± 0.0086	0.4566 ± 0.0282	0.6795 ± 0.0283	0.2622 ± 0.0318	0.3186 ± 0.0273
0.3519 ± 0.0209	**0.6900 ± 0.0185**	**0.4611 ± 0.0219**	0.6821 ± 0.0155	**0.2926 ± 0.0429**	**0.3230 ± 0.0189**
Sofa	Train	TVmonitor			Mean
0.1910	0.6170	0.2360			0.3660
0.2064 ± 0.0423	0.6568 ± 0.0219	0.3012 ± 0.0348			0.4032 ± 0.0026
0.2110 ± 0.0297	0.6682 ± 0.0207	0.3159 ± 0.0390			0.4181 ± 0.0055
0.1965 ± 0.0566	**0.6939 ± 0.0085**	**0.3520 ± 0.0402**			**0.4321 ± 0.0070**

Table 2
AP scores for all the classes using 50 positive and 50 negative labeled examples for each class on the MIR Flickr dataset.

Methods	Animals	Baby	Baby*	Bird	Bird*
MKL+LSR [11]	0.3100	0.0750	0.1610	0.1240	0.1630
GLP+SVR	0.3246 ± 0.0250	0.1286 ± 0.0290	**0.1899 ± 0.0127**	0.1573 ± 0.0276	0.1965 ± 0.0169
TR+GLP+SVR	0.3418 ± 0.0246	**0.1300 ± 0.0278**	0.1879 ± 0.0092	0.1658 ± 0.0162	**0.2038 ± 0.0175**
TR+GLP*+SVR	**0.3608 ± 0.0237**	0.1267 ± 0.0364	0.1892 ± 0.0030	**0.1731 ± 0.0059**	0.2008 ± 0.0277
Car	Car*	Clouds	Clouds*	Dog	Dog*
0.2290	0.3050	0.6120	0.5370	0.1820	0.2120
0.2510 ± 0.0231	0.4163 ± 0.0310	0.6169 ± 0.0345	0.5421 ± 0.0187	0.2498 ± 0.0074	0.2721 ± 0.0158
0.2722 ± 0.0259	0.4470 ± 0.0155	**0.6354 ± 0.0308**	**0.5568 ± 0.0210**	0.2597 ± 0.0075	0.2796 ± 0.0113
0.2886 ± 0.0259	**0.4614 ± 0.0173**	0.6260 ± 0.0293	0.5500 ± 0.0192	**0.2639 ± 0.0093**	**0.2941 ± 0.0158**
Female	Female*	Flower	Flower*	Food	Indoor
0.4400	0.3130	0.3730	0.4240	0.3330	0.5140
0.4300 ± 0.0166	0.3650 ± 0.0451	0.4245 ± 0.0069	0.5087 ± 0.0163	0.3914 ± 0.0258	0.5684 ± 0.0203
0.4332 ± 0.0151	0.3829 ± 0.0382	0.4391 ± 0.0098	0.5229 ± 0.0142	0.4064 ± 0.0227	0.5714 ± 0.0219
0.4255 ± 0.0186	**0.4121 ± 0.0322**	**0.4473 ± 0.0037**	**0.5290 ± 0.0185**	**0.4280 ± 0.0079**	**0.5817 ± 0.0253**
Lake	Male	Male*	Night	Night*	People
0.1590	0.3660	0.2550	0.4710	0.3680	0.6290
0.2148 ± 0.0237	0.3834 ± 0.0233	0.2703 ± 0.0433	0.5144 ± 0.0278	0.3837 ± 0.0327	0.6412 ± 0.0193
0.2212 ± 0.0207	**0.3836 ± 0.0173**	0.2860 ± 0.0324	**0.5197 ± 0.0264**	**0.4292 ± 0.0266**	0.6465 ± 0.0189
0.2242 ± 0.0309	0.3756 ± 0.0369	**0.2999 ± 0.0236**	0.5150 ± 0.0340	0.4181 ± 0.0092	**0.6507 ± 0.0204**
People*	Plant life	Portrait	Portrait*	River	River*
0.5540	**0.6130**	**0.4740**	0.4290	**0.2340**	0.0470
0.5489 ± 0.0237	0.6113 ± 0.0247	0.4160 ± 0.0423	0.4080 ± 0.0613	0.1897 ± 0.0365	**0.0872 ± 0.0089**
0.5569 ± 0.0143	0.6108 ± 0.0174	0.4424 ± 0.0461	0.4318 ± 0.0626	0.2051 ± 0.0297	0.0781 ± 0.0090
0.5607 ± 0.0118	0.5998 ± 0.0127	0.4368 ± 0.0125	**0.4777 ± 0.0462**	0.2236 ± 0.0166	0.0754 ± 0.0131
Sea	Sea*	Sky	Structures	Sunset	Transport
0.4370	0.2550	**0.6930**	**0.6550**	0.5430	**0.3210**
0.4328 ± 0.0229	0.2719 ± 0.0243	0.6839 ± 0.0466	0.6198 ± 0.0265	0.5533 ± 0.0236	0.2883 ± 0.0360
0.4471 ± 0.0187	**0.2768 ± 0.0162**	0.6920 ± 0.0499	0.6337 ± 0.0218	0.5639 ± 0.0248	0.2945 ± 0.0418
0.4453 ± 0.0204	0.2722 ± 0.0189	0.6925 ± 0.0575	0.6336 ± 0.0113	**0.5734 ± 0.0095**	0.3133 ± 0.0290
Tree	Tree*	Water			Mean
0.4530	0.2310	0.4520			0.3670
0.4269 ± 0.0276	0.2881 ± 0.0651	0.4534 ± 0.0312			0.3874 ± 0.0038
0.4165 ± 0.0276	0.3334 ± 0.0434	0.4526 ± 0.0287			0.3989 ± 0.0024
0.4085 ± 0.0246	**0.3511 ± 0.0397**	**0.4543 ± 0.0294**			**0.4042 ± 0.0022**

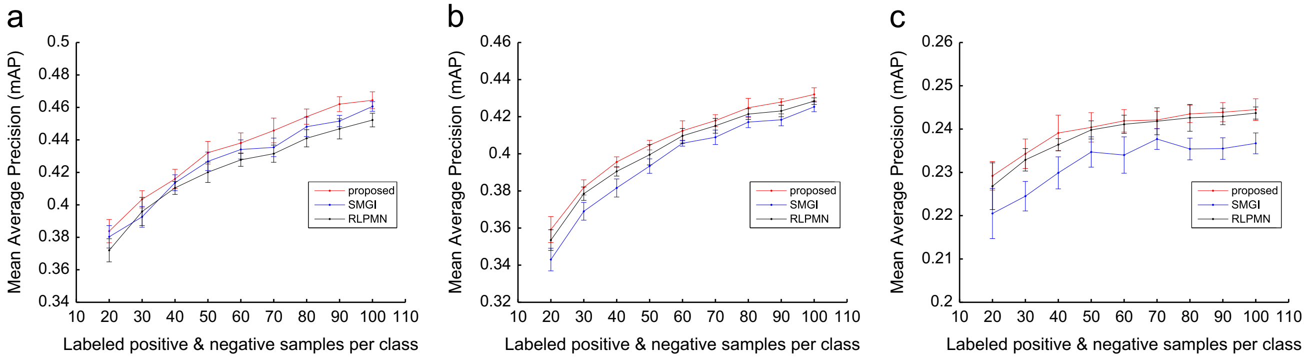


Fig. 5. Performance of different graph combination approaches in mAP (mean ± standard deviation) in the multimodal semi-supervised image classification task on the three datasets using varied number of labeled examples for each class. (a) PASCAL VOC'07. (b) MIR Flickr. (c) NUS-WIDE-Object.

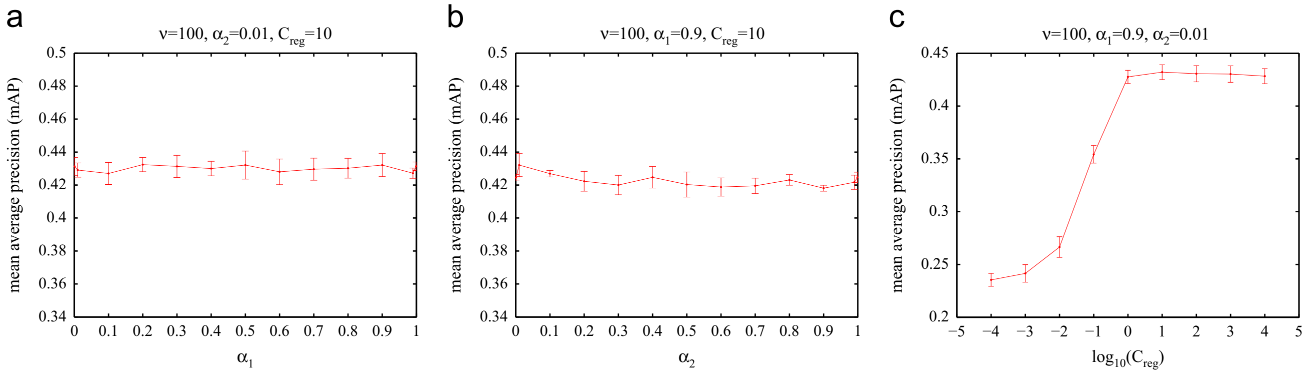


Fig. 6. Performance of *TR*+*GLP**+*SVR* in mAP (mean ± standard deviation) using 50 positive and 50 negative labeled examples for each class on the PASCAL VOC'07 dataset with varied: (a) α_1, (b) α_2, (c) C_{reg}.

SMGI [37] and RLPMN [30]. SMGI [37] is proposed to handle sparse integration of tens or hundreds of graphs by discarding irrelevant graphs. However, all the three graphs are relevant to the current task and thus sparsity is an inappropriate constraint. Therefore, SMGI performs unsatisfactorily here.

Besides, we observe in our experiments that RLPMN [30] converges quickly and the learned weights of the three graphs are always the same. Therefore, the results of RLPMN are similar to those of an average combination of the three graphs (i.e., *TR*+*GLP*+*SVR*), and thus the proposed graph combination approach in this paper performs better.

6.5. Hyperparameter tuning

At the beginning of this subsection, it should be noted that, due to the lack of labeled samples, a cross-validation strategy may not be applicable. Recall that there are four tunable hyperparameters in total in our model, i.e., α_1, α_2, ν and C_{reg}. In this subsection, we focus on the setting of these hyperparameters.

To begin with, the results of *TR*+*GLP**+*SVR* on the PASCAL VOC'07 dataset with 50 positive and 50 negative labeled examples are taken as examples. We report the mAP scores with varied α_1, α_2, and C_{reg} in Fig. 6. Note that α_1 and α_2 are chosen from $\{0.001, 0.01, 0.1, 0.2, \ldots, 0.9, 0.99, 0.999\}$, and C_{reg} is chosen from $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$. As shown in Fig. 6, α_1 is relatively insensitive, whereas a small α_2 is beneficial for the final performance. More importantly, a small C_{reg} may lead to catastrophic results, and thus a large C_{reg} (i.e., larger than 1) is necessary. We have also conducted experiments on other datasets and observed similar trends. Therefore, we adopt the same settings for the three aforementioned hyperparameters in all the experiments for conciseness and fairness, i.e., $\alpha_1 = 0.9$, $\alpha_2 = 0.01$ and $C_{reg} = 10$.

Table 3
Weights of different graphs and the mAP scores (mean ± standard deviation) of *TR*+*GLP**+*SVR* with varied ν using 50 positive and 50 negative labeled examples for each class on the PASCAL VOC'07 dataset.

ν	w_{tr}^{ν}	w_{tr}^{t}	w_{tr*}^{t}	mAP
1	0.0089	0.9857	0.0055	0.4284 ± 0.0063
2	0.0089	0.9856	0.0055	0.4282 ± 0.0040
5	0.0092	0.9851	0.0057	0.4304 ± 0.0048
10	0.0095	0.9846	0.0059	0.4307 ± 0.0053
20	0.0104	0.9831	0.0065	0.4318 ± 0.0061
50	0.0128	0.9792	0.0080	0.4291 ± 0.0044
100	0.0203	0.9668	0.0129	0.4321 ± 0.0070
200	0.2220	0.6095	0.1685	0.4236 ± 0.0058
500	0.3095	0.4151	0.2755	0.4219 ± 0.0072
1000	0.3236	0.3718	0.3046	0.4215 ± 0.0071
2000	0.3291	0.3520	0.3189	0.4192 ± 0.0043
5000	0.3317	0.3407	0.3276	0.4209 ± 0.0056
10,000	0.3325	0.3370	0.3305	0.4189 ± 0.0069
∞	0.3333	0.3333	0.3333	0.4182 ± 0.0083

As a next step, we investigate the tuning of ν. We observe in our experiments that the hyperparameter ν does not affect the performance too much when it is chosen to be relatively small, and the graph combination approach degenerates to a simple average combination when ν is chosen to be relatively large. Table 3 shows the weights of different graphs and the mAP scores with varied ν.

Recall that w_{tr}^{ν}, w_{tr}^{t} and w_{tr*}^{t} respectively denote the combination weights of visual graph, tag graph and refined tag graph, as shown in Eq. (6). From Table 3, we can observe that the mAP score is relatively insensitive to ν when ν is chosen to be small (i.e., no larger than 100). However, as ν becomes larger, the graph weights tend to be equal to each other and thus the approach to combining

Table 4
Running time (measured in seconds) on the three datasets of the following four approaches: *MKL+LSR*, *GLP+SVR*, *TR+GLP+SVR* and *TR+GLP*+SVR*.

Methods	PASCAL VOC'07	MIR Flickr	NUS-WIDE-Object
MKL+LSR [11]	531	7467	23,147
GLP+SVR(ours)	87	1525	4705
TR+GLP+SVR(ours)	125	1812	5581
TR+GLP*+SVR(ours)	257	3809	11,442

Table 5
Running time (measured in seconds) on the three datasets of the following three approaches in the multimodal semi-supervised image classification task: the proposed graph combination approach, SMGI [37] and RLPMN [30].

Methods	PASCAL VOC'07	MIR Flickr	NUS-WIDE-Object
Proposed	257	3809	11,442
SMGI [37]	384	9353	23,656
RLPMN [30]	382	9255	23,142

multiple graphs degenerates to a simple average combination. Therefore, we choose a small ν for all the experiments.

6.6. *Complexity issues*

Recall that we have compared four approaches in Section 6.3 to evaluate the effectiveness of each component of the proposed GraMSIC framework. To systematically investigate the complexity issues, we report in Table 4 the running time (measured in seconds) of the four approaches on the three datasets.

Note that we run MATLAB codes on a server with 2.20 GHz[3] CPU and 128 GB RAM. Among the four approaches, *GLP+SVR* is the most efficient, since only label propagation and SVR are involved. *TR+GLP+SVR* takes a little bit more time than *GLP+SVR* since the tag refinement procedure is integrated. *TR+GLP*+SVR* requires almost twice as much time as *TR+GLP+SVR* due to a few iterations of the EM algorithm. However, despite that the total computational complexity of the aforementioned three approaches is $O(n^2)$, the LSR has a time complexity of $O(n^3)$ due to the SVD of the centered visual kernel matrix. As a consequence, the proposed GraMSIC framework (i.e., *TR+GLP*+SVR*) performs more efficiently than the method in [11].

Moreover, we have also conducted experiments on the three datasets to evaluate the complexity of different approaches to graph combination. Concretely, we list in Table 5 the running time (measured in seconds) of the three approaches compared in Section 6.4. Note that all the three methods are used in the multimodal semi-supervised image classification task.

We can observe from Table 5 that the proposed graph combination approach performs more efficiently than the other two methods. This is due to the fact that SMGI [37] and RLPMN [30] are both proposed to handle general binary classification problems. However, there are many classes in the three datasets. For example, there are 20 classes in total in the PASCAL VOC'07 dataset, which means that there are 20 binary classification tasks in total.[4] SMGI and RLPMN learn to combine multiple graphs for each class separately, and thus require more time. In contrast to the aforementioned two methods, the proposed graph combination approach in this paper can learn graph combination weights for all the classes simultaneously, and thus is more efficient.

7. Conclusion

In this paper, we investigate an important task for image search engine on photo sharing websites, where training images come along with tags, but only a subset being labeled, and the goal is to infer the class label of test images without tags. We propose a GraMSIC framework to handle the task, which is made up of the following three components: (1) tag refinement is used to refine the inaccurate and incomplete tags on photo sharing websites such as Flickr; (2) graph-based label propagation is adopted to learn with a limited number of labeled samples, where the performance can be further enhanced by using the proposed approach to combining multiple graphs; (3) SVR is adopted to predict the class label of test images by readily leveraging the image features in the RKHS. Experimental results show that the proposed method performs more efficiently and achieves significantly better results than existing methods.

Acknowledgments

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China under Grant 2012AA012503, National Key Technology Research and Development Program of China under Grant 2012BAH07B01, Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120001110097, and National Natural Science Foundation of China under Grants 61371128 and 61073084.

References

[1] R. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. (6) (1973) 610–621.
[2] A. Khotanzad, Y. Hong, Invariant image recognition by Zernike moments, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 489–497.
[3] Z. Hong, Algebraic feature extraction of image for recognition, Pattern Recognit. 24 (3) (1991) 211–219.
[4] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Trans. Neural Netw. 10 (5) (1999) 1055–1064.
[5] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, in: IEEE International Conference on Computer Vision (ICCV), 2007, pp. 1–8.
[6] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[7] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2006, pp. 2169–2178.
[8] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[9] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
[10] X. Zhu, Semi-supervised learning literature survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
[11] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 902–909.
[12] F. Bach, G. Lanckriet, M. Jordan, Multiple kernel learning, conic duality, and the smo algorithm, in: Proceedings of the International Conference on Machine Learning, 2004, p. 6.
[13] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, et al., SimpleMKL, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[14] A. Berlinet, C. Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics, Kluwer Academic Publisher, 2004.
[15] W. Xie, Z. Lu, Y. Peng, J. Xiao, Multimodal semi-supervised image classification by combining tag refinement, graph-based learning and support vector regression, in: IEEE International Conference on Image Processing, 2013, pp. 4307–4311.
[16] M. Huiskes, M. Lew, The MIR Flickr retrieval evaluation, in: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, 2008, pp. 39–43.
[17] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from National University of Singapore, in: Proceedings of the ACM International Conference on Image and Video Retrieval, 2009, pp. 1–9.
[18] S.Z. Li, Markov Random Field Modeling in Image Analysis, Springer, 2009.
[19] X. Li, C. Snoek, M. Worring, Learning social tag relevance by neighbor voting, IEEE Trans. Multimed. 11 (7) (2009) 1310–1322.

[3] We perform all the experiments with only a single thread.

[4] 1-vs-all strategy is adopted in all the experiments since a single image may contain multiple class labels.

[20] D. Liu, X. Hua, L. Yang, M. Wang, H. Zhang, Tag ranking, in: Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 351–360.
[21] L. Chen, D. Xu, I. Tsang, J. Luo, Tag-based web photo retrieval improved by batch mode re-tagging, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3440–3446.
[22] H. Xu, J. Wang, X. Hua, S. Li, Tag refinement by regularized lda, in: Proceedings of the 17th ACM International Conference on Multimedia, 2009, pp. 573–576.
[23] G. Zhu, S. Yan, Y. Ma, Image tag refinement towards low-rank, content-tag prior and error sparsity, in: Proceedings of the International Conference on Multimedia, 2010, pp. 461–470.
[24] D. Liu, X. Hua, M. Wang, H. Zhang, Image retagging, in: Proceedings of the International Conference on Multimedia, 2010, pp. 491–500.
[25] G. Miller, C. Fellbaum, Wordnet: An Electronic Lexical Database, 1998.
[26] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, vol. 16, 2004, pp. 321–328.
[27] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 912–919.
[28] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[29] K. Tsuda, H. Shin, B. Scholkopf, Fast protein classification with multiple networks, Bioinformatics 21 (Suppl. 2) (2005) ii59–ii65.
[30] T. Kato, H. Kashima, M. Sugiyama, Robust label propagation on multiple networks, IEEE Trans. Neural Netw. 20 (1) (2009) 35–44.
[31] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, Q. Morris, et al., Genemania: a real-time multiple association network integration algorithm for predicting gene function, Genome Biol. 9 (2008) S4–S15.
[32] S. Mostafavi, Q. Morris, Fast integration of heterogeneous data sources for predicting gene function with limited annotation, Bioinformatics 26 (14) (2010) 1759–1765.
[33] D. Warde-Farley, S.L. Donaldson, O. Comes, K. Zuberi, R. Badrawi, P. Chao, M. Franz, C. Grouios, F. Kazi, C.T. Lopes, et al., The genemania prediction server: biological network integration for gene prioritization and predicting gene function, Nucleic Acids Res. 38 (2010) W214–W220.
[34] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, J.S. Kandola, On kernel-target alignment, in: Advances in Neural Information Processing Systems, 2001, pp. 367–373.
[35] A. Argyriou, M. Herbster, M. Pontil, Combining graph Laplacians for semi-supervised learning, in: Advances in Neural Information Processing Systems, 2005, pp. 67–74.
[36] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Trans. Circuits Syst. Video Technol. 19 (5) (2009) 733–746.
[37] M. Karasuyama, H. Mamitsuka, Multiple graph label propagation by sparse integration, IEEE Trans. Neural Netw. Learn. Syst.
[38] T. Zhang, A. Popescul, B. Dom, Linear prediction models with graph regularization for web-page categorization, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 821–826.
[39] J. Verbeek, M. Guillaumin, T. Mensink, C. Schmid, Image annotation with tagprop on the mirflickr set, in: Proceedings of the International Conference on Multimedia Information Retrieval, 2010, pp. 537–546.
[40] C.M. Bishop, et al., Pattern Recognition and Machine Learning, springer, New York, 2006.
[41] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B (Methodological) (1977) 1–38.
[42] C. Chang, C. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[43] K. Zhu, H. Wang, H. Bai, J. Li, Z. Qiu, H. Cui, E.Y. Chang, Parallelizing support vector machines on distributed computers, in: Advances in Neural Information Processing Systems, 2007, pp. 257–264.
[44] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[45] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[46] Y. Luo, D. Tao, B. Geng, C. Xu, S. Maybank, Manifold regularized multi-task learning for semi-supervised multi-label image classification, IEEE Trans. Image Process. 22 (2) (2013) 523–536.

Wenxuan Xie received the B.Sc. degree from Nanjing University, Nanjing, China, in 2010. He is currently pursuing the Ph.D. degree with the Institute of Computer Science and Technology, Peking University, Beijing, China. His current research interests include computer vision, machine learning, and social media analysis.

Zhiwu Lu received the M.Sc. degree in applied mathematics from Peking University, Beijing, China, in 2005, and the Ph.D. degree in computer science from City University of Hong Kong, in 2011. Since March 2011, he has become an assistant professor with the Institute of Computer Science and Technology, Peking University. He has published over 30 papers in international journals and conference proceedings including TIP, TSMC-B, TMM, AAAI, ICCV, CVPR, ECCV, and ACM-MM. His research interests lie in machine learning, computer vision, and multimedia information retrieval.

Yuxin Peng is the professor and director of Multimedia Information Processing Lab (MIPL) in the Institute of Computer Science and Technology (ICST), Peking University. He received the Ph.D. degree in computer application from School of Electronics Engineering and Computer Science (EECS), Peking University, in July 2003. After that he worked as an assistant professor in ICST, Peking University. From August 2003 to November 2004, he was a visiting scholar with the Department of Computer Science, City University of Hong Kong. He was promoted to associate professor and professor in Peking University in August 2005 and August 2010 respectively. In 2006, he was authorized by the "Program for New Star in Science and Technology of Beijing", and the "Program for New Century Excellent Talents in University (NCET)". He has published over 60 papers in refereed international journals and conference proceedings including IJCV, TCSVT, TMM, TIP, PR, ACM-MM, ICCV, CVPR, AAAI, IJCAI and ICDM. He led his team to participate in TRECVID (TREC Video Retrieval Evaluation). In TRECVID 2009, his team won four first places on 4 sub-tasks and two second places on the left 2 sub-tasks in the High-Level Feature Extraction (HLFE) task and Search task. In TRECVID 2012, his team gained three first places on 3 sub-tasks and one second places on the left 1 sub-task in the Known-Item Search (KIS) task and Instance Search (INS) task. Besides, he has obtained 13 patents. His current research interests mainly include video and image understanding and retrieval, and multimedia search and mining.

Jianguo Xiao is the professor and head in the Institute of Computer Science and Technology (ICST), Peking University, Beijing, China. He received his M.S. degree in computer science and technology from Peking University, in 1988. His research interests mainly include image and video processing, and text mining. He has published over 50 papers in refereed international journals and conference proceedings. For his work and contributions, he was the recipient of some famous awards in China, including the first prize of the National S&T Progress Award in 1995, the second prize of the National S&T Progress Award in 2006, 2007 and 2009. In 2008, he won the 7th Guanghua Award of Engineering. In 2010, he gained the 10th Bisheng Award for Outstanding Achievement in Printing.