Optimal Graph Learning with Partial Tags and Multiple Features for Image and Video Annotation

Lianli Gao[1], Jingkuan Song[2], Feiping Nie[3], Yan Yan[2], Nicu Sebe[2], Heng Tao Shen[4]
[1]University of Electronic Science and Technology of China, China.
lianli.gao@uestc.edu.cn

[2]University of Trento, Italy
{jingkuan.song,yan.yan,niculae.sebe}@unitn.it

[3]University of Texas, Arlington
feipingnie@gmail.com

[4]The University of Queensland, Australia
shenht@itee.uq.edu.au

Abstract

In multimedia annotation, due to the time constraints and the tediousness of manual tagging, it is quite common to utilize both tagged and untagged data to improve the performance of supervised learning when only limited tagged training data are available. This is often done by adding a geometrically based regularization term in the objective function of a supervised learning model. In this case, a similarity graph is indispensable to exploit the geometrical relationships among the training data points, and the graph construction scheme essentially determines the performance of these graph-based learning algorithms. However, most of the existing works construct the graph empirically and are usually based on a single feature without using the label information. In this paper, we propose a semi-supervised annotation approach by learning an optimal graph (OGL) from multi-cues (i.e., partial tags and multiple features) which can more accurately embed the relationships among the data points. We further extend our model to address out-of-sample and noisy label issues. Extensive experiments on four public datasets show the consistent superiority of OGL over state-of-the-art methods by up to 12% *in terms of mean average precision.*

1. Introduction

Recently, we have witnessed an exponential growth of user generated videos and images, due to the booming of social networks, such as Facebook and Flickr. Consequently, there are increasing demands to effectively organize and access these multimedia data via tagging. One promising direction is to combine labeled data (often of limited amount) and a huge pool of unlabeled data in forming abundant training resources for optimizing annotation models, which is referred to semi-supervised learning (SSL). This is often done by adding a geometrically based regularization term in the objective function of a supervised learning model. To exploit the geometrical relationships among the training data points, the algorithms treat both labeled and unlabeled samples as vertices (nodes) in a graph and build pairwise edges between these vertices which are weighed by the affinities (similarities) between the corresponding sample pairs.

SSL has been widely studied [3, 28, 13, 32, 4] and applied to many challenging tasks [4, 2, 3] such as image annotation and image retrieval. By exploiting a large number of unlabeled data with reasonable assumptions, SSL can reduce the need of expensive labeled data and thus achieve promising results especially for noisy labels [25]. The harmonic function approach [35] and local and global consistency (LGC) [34] are two representative graph-based SSL methods. The harmonic function approach [35] emphasizes the harmonic nature of the energy function and LGC considers the spread of label information in an iterative way. While these two methods are transductive, manifold regularization (MR) [1, 21] is inductive. In practice, MR extends regression and SVM to semi-supervised learning methods such as Laplacian Regularized Least Squares (LapRLS) and Laplacian Support Vector Machines (LapSVM) respectively by adding a geometrically based regularization term [17].

Recently, many applications [33, 25, 24] were proposed. Zhang *et al.* [33] extended LDA to semi-supervised discriminant analysis, Tang *et al.* [25] addressed the noisy label issue for the task of semi-supervised image labeling, and Song *et al.* [24] utilized weak-label information for cross-media retrieval.

Since an informative graph is critical for the graph-based algorithms, its construction has been extensively studied [27, 3, 18, 22, 30, 23]. The most popular way to construct a graph is the K-nearest neighbor (or ϵ-range neighbor) method, where, for each data point, the samples are connected with its K-nearest neighbors (or ϵ-range-neighbor). Then the Gaussian-kernel can be used to quantify the graphs. However, the tuning of σ in the Gaussian-kernel approach is empirical [27]. Recently, it has been proposed to learn the graph by considering the pairwise distance-based and the reconstruction coefficients-based methods. The former is based on the assumption that close data points should have a high similarity and vice versa. The latter assumes that each data point can be reconstructed as a linear combination of the other data points. These two methods show different strengths and weaknesses in various applications. However, most of these graphs are constructed on a single information cue (e.g., visual feature, labels), and an optimal graph utilizing multiple cues has rarely been addressed.

To address the above issues, in this work, we propose learning an optimal graph (OGL) from multi-cues (i.e., partial tags and multiple features), which can more accurately encode the relationships between data points. Then, we incorporate the learned optimal graph with the SSL model, and we further extend this model to address out-of-sample extension and noisy label issues. It is worthwhile to highlight the following aspects of the proposed approach here:

- As far as we know, this is the first work to explicitly learn an optimal graph both from labels and multiple features, and we propose an efficient way to solve the optimization problem. The learned optimal graph can automatically determine the confidence of the partial tags and different visual features to more precisely reflect the relationships among data points.

- Our optimal graph learning method is a general framework, and theoretically, it can be incorporated with other graph-based learning methods. Specifically, we integrate OGL with the SSL method, and evaluate our OGL for image and video annotation.

- We further discuss and propose solutions for out-of-sample extension, noisy label issues and different graph construction methods for our models, to deal with the real applications.

- Experiments for image and video annotation on real world datasets demonstrate the superiority of OGL over existing graph construction methods.

2. Related Work

There are generally two ways to build a similarity graph. One is based on pairwise distances (e.g., Euclidean distance), and the other is based on reconstruction coefficients. The first method is based on a reasonable assumption that close data points should have a high similarity and vice versa. The second method assumes that each data point could be represented as a linear combination of the other points. When the data are clean, i.e., the data points are strictly sampled from the subspaces, several approaches are able to recover the subspaces [16]. However, in real applications, the data set may lie at the union of multiple subspaces or contain noise and outliers [18]. As a result, inter-class data points may be connected with very high weights. Hence, eliminating the effects of errors becomes a major challenge. To address these problems, several algorithms have been proposed, e.g., Locally Linear Manifold Clustering (LLMC) [7], Agglomerative Lossy Compression (ALC) [19], Sparse Subspace Clustering (SSC) [5], L1-graph [3, 29], Low Rank Representation (LRR) [10, 9], Latent Low Rank Representation (LatLRR) [11], Fixed Rank Representation (FRR) [12], L2Graph [18] (please refer to [26] for a comprehensive survey on these algorithms).

Of the above methods, SSC [5] and L1-graph [3] obtain a sparse similarity graph from the sparsest coefficients. One of the main differences between these two techniques is that [5] formulates the noise and outliers in the objective function and provides more theoretical analysis, whereas [3] derives a series of algorithms upon the L1-graph for various tasks. The popular LRR model [10, 9] and its extensions [11, 12] are very similar to SSC, except that they aim to obtain a similarity graph from the lowest-rank representation rather than the sparsest one. Both ℓ_1 and rank-minimization-based methods can automatically select neighbors for each data point by adopting the sparse solution, and have achieved impressive results in numerous applications. However, their computational complexity is proportional to the cube of the problem size. Moreover, SSC requires that the corruption over each data point has a sparse structure, and LRR assumes that only a small portion of the data is contaminated, otherwise the performance will degrade. In fact, these two problems are mainly caused by the adopted error-handling strategy, i.e., removing the errors from the data set to obtain a clean dictionary over which each sample is encoded [18].

Differently, our OGL (described in the next section) eliminates the effects of errors and builds a more precise graph by considering both partial tags and multiple features, and automatically assigning a higher confidence to good performed graphs constructed on each information cue.

3. Our Approach

In this section, we introduce our method OGL which consists of two phases (see Fig. 1). Firstly, a similarity graph is constructed on each feature (multiple feature graph) and also on the partial tags (partial label graph) to exploit the relationship among the data points. Partial tags

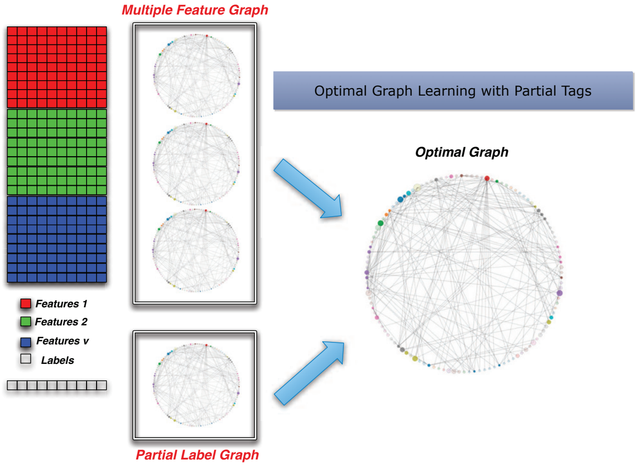


Figure 1. The overview of OGL.

means that tags are provided only for a part of the training data. Then, the optimal graph learning is applied to these graphs to construct an optimal graph, which is integrated with SSL for the task of image and video annotation. Note that in theory our approach can be integrated with all kinds of graph-based algorithms.

3.1. Terms and Notations

We first introduce the notations which will be used in the rest of the paper. $X = \{x_1, x_2, ..., x_n\}$ represents a set of n images, and $y_i = \{0, 1\}^c$ is the label for the i-th image ($1 \leq i \leq n$), and c is the number of annotations/classes. The first l points x_i ($i \leq l$) are labeled as Y_l, and the remaining u points x_i ($l + 1 \leq i \leq n$) are unlabeled. The goal of transductive graph-based SSL is to predict the label F_u of the unlabeled points. Define a $n \times c$ matrix $F = \begin{bmatrix} F_l \\ F_u \end{bmatrix}$ with $F_l = Y_l$ and $F_u = \{0\}^{u*c}$.

Suppose that for each image, we have v features. Let $X^t = \{x_i^t\}_{i=1}^n$ denote the feature matrix of the t-th view of training images, where $t \in \{1, ..., v\}$.

3.2. Optimal graph-based SSL

The traditional graph based semi-supervised learning usually solves the following problem:

$$\min_{F, F_l = Y_l} \sum_{ij} \|f_i - f_j\|_2^2 s_{ij} \tag{1}$$

where f_i and f_j are the labels for the i-th and j-th images, and S is the affinity graph with each entry s_{ij} representing the similarity between two images. The affinity graph $S \in \mathbb{R}^{n \times n}$ is usually defined as follows:

$$s_{ij} = \begin{cases} e^{-\|x_i - x_j\|_2^2 / 2\sigma^2}, \text{ if } x_i \in \mathcal{N}_K(x_j) \text{ or } x_j \in \mathcal{N}_K(x_i) \\ 0, \text{else} \end{cases} \tag{2}$$

where $\mathcal{N}_K(\cdot)$ is the K-nearest neighbor set and $1 \leq (i, j) \leq n$. The variance σ will affect the performance significantly, and it is usually empirically tuned. Also, the similarity graph is often derived from single information cue. To address these issues, we propose to learn an optimal graph S from multiple cues.

The multiple cues include the given label information F and the multiple features $X^t = \{x_i^t\}_{i=1}^n$. An optimal graph S should be smooth on all these information cues, which can be formulated as:

$$\min_{S, \alpha} g(F, S) + \mu \sum_{t=1}^{v} \alpha^t h\left(X^t, S\right) + \beta r(S, \alpha) \tag{3}$$

where $g(F, S)$ is the penalty function to measure the smoothness of S on the label information F and $h(X^t, S)$ is the loss function to measure the smoothness of S on the feature X^t. $r(S, \alpha)$ are regularizers defined on the target S and α. μ and β are balancing parameters, and α^t determines the importance of each feature.

The penalty function $g(F, S)$ should be defined in such a way that close labels have high similarity and vice versa. In this paper, we define it as follows:

$$g(F, S) = \sum_{ij} \|f_i - f_j\|_2^2 s_{ij} \tag{4}$$

where f_i and f_j are the labels of data points x_i and x_j. Similarly, $h(X^t, S)$ can be defined as:

$$h\left(X^t, S\right) = \sum_{ij} \left\|x_i^t - x_j^t\right\|_2^2 s_{ij} \tag{5}$$

Note that for simplicity, we use distance based method to learn the similarity graph here. Other options based on the reconstruction coefficients methods can be utilized to achieve better performance, which is discussed in the next section. Instead of preserving all the pairwise distances, we consider preserving the pair distances of the K-nearest neighbors here, i.e., if x_i^t and x_j^t (or f_i *and* f_j) are not K-nearest neighbors of each other, their distance will be set to a large constant. The regularizer term $r(S, \alpha)$ is defined as:

$$r(S, \alpha) = \frac{\mu\gamma}{\beta} \|S\|_F^2 + \|\alpha\|_2^2 \tag{6}$$

We further constrain that $S \geq 0$, $S\mathbf{1} = \mathbf{1}$, $\alpha \geq 0$ and $\alpha^T\mathbf{1} = \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is a column vector with all ones. Then we can obtain the objective function for learning the optimal graph by replacing $g(F, S)$, $h(X^t, S)$ and $r(S, \alpha)$ in Eq.3 using Eq.4, Eq.5 and Eq.6. By combining Eq.1 with Eq.3, we can obtain the objective function for optimal-graph based SSL, as follows:

$$\begin{array}{l} \min\limits_{S,F,\alpha} \sum\limits_{ij} \|f_i - f_j\|_2^2 s_{ij} + \beta \|\alpha\|_2^2 \\ +\mu \left(\sum\limits_{tij} \left(\alpha_t \left\|x_i^t - x_j^t\right\|_2^2 s_{ij} \right) + \gamma \|S\|_F^2 \right) \\ s.t. \left\{ \begin{array}{c} S \geq 0, S\mathbf{1} = \mathbf{1} \\ F_l = Y_l \\ \alpha \geq 0, \alpha^T \mathbf{1} = \mathbf{1} \end{array} \right. \end{array} \tag{7}$$

3.3. Iterative optimization

We propose an iterative method to minimize the above objective function in Eq.7. Firstly, we initialize $S = \sum_t S^t/v$ with each S^t being calculated using Eq.2, and we initialize $\alpha^t = 1/v$. We further normalize S as $S = (D^{1/2})^T S D^{1/2}$. Once these initial values are given, in each iteration, we first update F given S and α, and then update S and α by fixing the other parameters. These steps are described below.

3.3.1 Update F

By fixing S and α, we can obtain F by optimizing Eq.7. This is equivalent to optimize the following objective function:

$$\min_{F,F_l=Y_l} \sum_{ij} \|f_i - f_j\|_2^2 s_{ij} = \min_{F,F_l=Y_l} \left\|F\,(I-S)\,F^T\right\|_F^2 \tag{8}$$

where I is an identity matrix. Let $L = I - S$, and differentiate the objective function in Eq.8 with respect to F, we obtain:

$$\begin{array}{l} LF = 0 \Rightarrow \left[\begin{array}{cc} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{array} \right] \left[\begin{array}{c} F_l \\ F_u \end{array} \right] = 0 \\ \Rightarrow \left\{ \begin{array}{l} L_{ll}F_l + L_{lu}F_u = 0 \\ L_{ul}F_l + L_{uu}F_u = 0 \end{array} \right. \end{array} \tag{9}$$

Then we can obtain:

$$F_u^* = -{L_{uu}}^{-1} L_{ul} F_l \tag{10}$$

3.3.2 Update S

By fixing F and α, we can obtain S by optimizing Eq.7. It is equivalent to optimize the following objective function:

$$\sum_{ij} \|f_i - f_j\|_2^2 s_{ij} + \mu \sum_{tij} \left(\alpha_t \left\|x_i^t - x_j^t\right\|_2^2 s_{ij} \right) \tag{11}$$
$$+\mu\gamma \|S\|_F^2$$

It can be reformulated as:

$$\begin{array}{l} \min\limits_{S,S\geq 0,S1=1} \sum\limits_i tr\left(\mu\gamma s_i {s_i}^T + (a_i + \mu b_i)\, {s_i}^T\right) \\ \Rightarrow \min\limits_{S,S\geq 0,S1=1} \sum\limits_i tr\left(s_i {s_i}^T + \frac{a_i+\mu b_i}{\mu\gamma} {s_i}^T\right) \end{array} \tag{12}$$

and it is equivalent to:

$$\min_{S,S\geq 0,S1=1} \sum_i \left\| s_i + \frac{a_i + \mu b_i}{2\mu\gamma} \right\|_2^2 \tag{13}$$

where $b_i = \{b_{ij}, 1 \leq j \leq n\}$ with $b_{ij} = \sum\limits_t \alpha_t \left\|x_i^t - x_j^t\right\|_2^2$ and $a_i = \{a_{ij}, 1 \leq j \leq n\} \in R^{1\times n}$ with $a_{ij} = \|y_i - y_j\|_2^2$.

The problem in Eq.13 is simplex and we use the accelerated projected gradient method to linearly solve this problem. The critical step of the projected gradient method is to solve the following proximal problem:

$$\min_{x\geq 0,x^T1=1} \frac{1}{2}\|x - c\|_2^2 \tag{14}$$

This proximal problem can be solved using KKT approach. Then each s_i can be efficiently solved, and we can get the updated graph S.

3.3.3 Update α

By fixing F and S, we can obtain α by optimizing Eq.7. It is equivalent to optimize the following objective function:

$$\begin{array}{l} \min\limits_{\alpha\geq 0,\alpha^T1=1} \mu \sum\limits_t \alpha_t \left(\sum\limits_{ij} \left\|x_i^t - x_j^t\right\|_2^2 s_{ij} \right) + \beta \|\alpha\|_2^2 \\ \Rightarrow \min\limits_{\alpha\geq 0,\alpha^T1=1} \mu d\alpha + \beta \|\alpha\|_2^2 \end{array} \tag{15}$$

where $d = \{d_t, 1 \leq t \leq v\}$ with $d_t = \sum\limits_{ij} \left\|x_i^t - x_j^t\right\|_2^2 s_{ij}$. It can be reformulated in the form of problem in Eq.14 and can be solved similarly to obtain α.

We update F, S and α iteratively until the objective function Eq.7 converges.

4. Extensions of OGL

In this section, we discuss several issues for graph based learning methods in real applications and provide the solutions.

4.1. Out-of-sample extension

Out-of-sample refers to learning an annotation function that is able to label new data points. This can be achieved by adding a fitting model and a regularizer to the objective function in Eq.7, e.g., $\|XW + \mathbf{1}b - F\|_F^2 + \eta \|W\|_F^2$, where $W \in \mathbb{R}^{m\times c}, b \in \mathbb{R}^{1\times c}$ and $\mathbf{1}$ is a vector of all ones. To obtain the optimal solution W^* and b^*, we set the derivatives of the objective function with respect to W and b equals to zero. We have:

$$b^* = \frac{1}{n}\left(\mathbf{1}^T F - \mathbf{1}^T XW\right) \tag{16}$$

$$W^* = \left(X^T L_c X + \eta I\right)^{-1} X^T L_c F \quad (17)$$

where X is the concatenation of different features X^t, and $L_c = I - \mathbf{1}\mathbf{1}^T$. Then $\|XW + \mathbf{1}b - F\|_F^2 + \eta \|W\|_F^2$ can be reformulated as:

$$tr(F^T B F) \quad (18)$$

where $B = L_c - L_c X \left(X^T L_c X + \eta I\right)^{-1} X^T L_c$. Then, by adding this fitting model, F can be obtained by solving:

$$\begin{aligned} &\min_F trF^T\left(I - S + \omega B\right)F + tr(F-Y)^T U\left(F - Y\right) \\ &\Rightarrow F^* = \left(I + U - S + \omega B\right)^{-1} UY \end{aligned} \quad (19)$$

where ω is the parameter for the fitting model. Then we can obtain the annotation function W and b. Note that other fitting models can also be applied here, e.g., SVM, fast image tagging [2]. In [2], Chen et al. address the incomplete tagging problem by introducing a term $\tilde{B}$ to enrich the existing tags. Then a co-regularized learning scheme is adopted to jointly learn the annotation function $\tilde{W}$ and tag enrich function $\tilde{B}$, as follows: $\sum_i \left\|\tilde{B}f_i - x_i\tilde{W}\right\|_2^2$. For simplicity, we utilize the least square regression model to tackle the out-of-sample problem. Moreover, the performance can be further improved by incorporating better fitting models.

4.2. Noisy labels

The user-provided tags may be noisy. To address this issue, instead of limiting that the predicted labels F_l are strictly equal to the given hard labels Y_l, we introduce a soft error term $\|F_l - Y_l\|_F^2$ to release this constraint. Then, by fixing S and α, F is obtained by solving:

$$\begin{aligned} &\min_F \sum_{ij} \|f_i - f_j\|_2^2 s_{ij} + \mu \|F_l - Y_l\|_F^2 \\ &= \min_F trF^T\left(I - S\right)F + tr(F-Y)^T U\left(F - Y\right) \end{aligned} \quad (20)$$

where $U \in \mathbb{R}^{n\times n}$ is a diagonal matrix. By setting the derivative of the Eq.20 *w.r.t* F to zero, we have:

$$\begin{aligned} &F^* - SF^* + U\left(F^* - Y\right) = 0 \\ &\Rightarrow F^* = (I + U - S)^{-1} UY \end{aligned} \quad (21)$$

where $Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}$ with $Y_u = \{0\}^{u*c}$. Experimental results show that Eq.21 has superior performance over Eq.10.

4.3. Different graph construction models

In our work, we utilize a distance-based method to construct the similarity graph in Eq.4 and Eq.5 for simplicity. OGL can be further extended by using different graph construction models. One possible way is to adopt the reconstruction coefficients methods, which can be calculated by solving:

$$\min_S \sum_i \|x_i - D_i s_i\| \;\; s.t \;\; \mathbf{1}^T s_i = 1 \quad (22)$$

where $s_i \in \mathbb{R}^{n\times 1}$ is the coefficient of x_i over D_i and D_i consists of the K-nearest neighbors of x_i in Euclidean space.

Recently, some studies have exploited the inherent sparsity of sparse representation to obtain a block-diagonal affinity matrix, e.g., SSC [5] and the L1-graph [3]. In [3], the L1-graph is proposed for image analysis to solve the following problem:

$$\min_S \|s_i\|_1 \;\; s.t. \;\; \|x_i - X_i s_i\|_2 < \delta \quad (23)$$

where $s_i \in \mathbb{R}^{n\times 1}$ is the sparse representation of x_i over the dictionary X_i and δ is the error tolerance.

Another recently proposed method, LRR [10, 9], aims to find the lowest-rank representation, rather than the sparsest, by solving:

$$\min_{S,E} \;\; rank\left(S\right) + \lambda\|E\|_{2,1}, \;\; s.t. \;\; X = XS + E \quad (24)$$

where $S \in \mathbb{R}^{n\times n}$ is the coefficient matrix of X over the data set itself and E is the reconstruction error. These graph construction methods have reported superiority over distance-based graph construction method. Our OGL can be further improved if these graph construction methods were adopted.

5. Experiments

We evaluate our algorithm on the task of image and video annotation. Firstly, we study the influence of the parameters in our algorithm. Then, we compare our results with state-of-the-art algorithms on four standard datasets.

5.1. Experimental settings

5.1.1 Datasets

We consider four publicly available datasets that have been widely used in previous work.

IXMAS. This video dataset consists of 12 action classes (e.g., check watch, cross arms and scratch head). Each action is executed three times by 12 actors and is recorded with five cameras observing the subjects from very different perspectives.

NUS-WIDE. This image dataset contains 269,648 images downloaded from Flickr. Tagging ground-truth for 81 semantic concepts is provided for evaluation. Similar to [14], we only use the images associated with the 10 most frequent concept annotations obtaining 167,577 images in total.

ESP Game. It contains a wide variety of images including logos, drawings, and personal photos. Following [2], we use the subset of 20,000, out of the 60,000 images publicly available.

IAPR TC12. This set of 20,000 images accompanied with descriptions in several languages was initially published for cross-lingual retrieval [2].

For IXMAS, we use half of the tagged data to infer the tags for the rest data. For the other three datasets, out-of-sample extension is utilized to annotate the test data points. We randomly select 10,000 data points for training, and use the rest for testing. In the training dataset, partial data points are with tags.

5.1.2 Features

For IXMAS, we extract 500-D BoW feature based on SIFT for each camera, and the 5 cameras are taken as 5 features, following [31]. For NUS-WIDE, six types of low-level features are provided, including 64-D color histogram, 144-D color correlogram, 73-D edge direction histogram, 128-D wavelet texture, 225-D block-wise color moments and 500-D BoW based on SIFT descriptions. For ESPGAME and IAPR TC12, we use the provided 15 different visual descriptors for each dataset, including one Gist descriptor, six global color histograms, and eight local BoW features.

5.1.3 Baseline methods and evaluation metrics

To evaluate the performance of the optimal graph, we compare OGL with different graph construction methods, e.g., LGC [34], LLE [20], L2graph [18]. We further extend and compare all these methods on larger datasets by out-of-sample extension. We also compare against TagProp [8] and fastTag [2] algorithms, which currently achieve the best performance on the benchmark datasets. To test different fitting models, we combine OGL with fastTag to see the performance. We also compare with MBRM [6] and JEC [15].

We evaluate our models with standard performance measures:

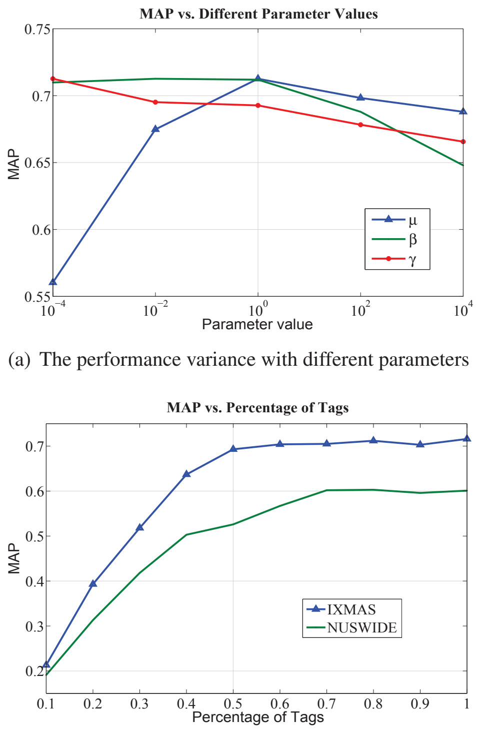
Precision vs recall over each label is calculated by computing the precision at each recall.

Mean average precision (MAP) over labels is calculated by computing for each label the average of the precisions measured after each relevant image is retrieved.

5.2. Parameters

There are several parameters, e.g., μ, β, γ and the number of tagged training data, affecting the performance of our algorithm. In this subsection, we study the performance variance with different parameters. Due to the space limit, we only report the results on the IXMAS dataset. The default settings for the parameters are: $\mu = 1$, $\beta = 1$, $\gamma = 10^{-4}$ and half of the training data points are with tags.

We tune μ, β and γ from $10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}$, and the results are shown in Fig. 2(a). When μ is relatively small, e.g., $\mu = 10^{-4}$, the worst performance is achieved. That is to say, if the impact of partial tags is dominant in constructing the optimal graph S, the MAP is unsatisfactory. With the increase of μ, the performance is slightly improved until reaching the peak at $\alpha = 10^{0}$. Further raising μ will result in a slight drop of the performance, which means that balancing the partial tags and multiple features is better than using only one of them. On the other hand, the performance keeps stable when β is relative small, e.g., $\beta \leq 10^{0}$, and it decreases quickly when β is set to a large value. Also, with the increase of γ, the performance drops.

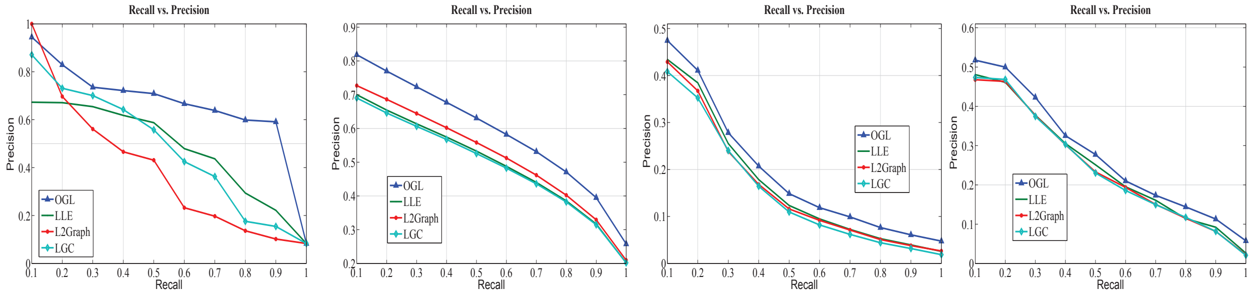


(a) The performance variance with different parameters

(b) The performance variance with different number of tagged training data

Figure 2. Experimental results under varying parameters

The size of tagged training dataset affects the accuracy of the model. We use 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the training data as tagged data and the rest as untagged data. The performance variances are illustrated in Fig. 2(b). There is an increasing trend with the rising of the number of tagged data. When 60% of the training data are tagged, further increasing the number of the tagged data will not improve the MAP significantly. From Fig. 2(b), it can be inferred that by using 30-40% of the tagged data of IXMAS, and 60-70% of NUS-WIDE, OGL can achieve comparable or even better results than other methods using all the tagged images (see the results in Table 1).



(a) Results on IXMAS (b) Results on NUS-WIDE (c) Results on ESP Game (d) Results on IAPR TC12

Figure 3. Experiment results on four datasets

Table 1. MAP Results on different datasets

Methods	IXMAS	NUS-WIDE	ESP Game	IAPR TC12
LGC	47.6	48.5	15.9	24.5
LLE	52.4	52.1	16.7	25.0
L2Graph	59.6	54.8	18.1	26.4
OGL	**71.3**	**60.2**	**20.2**	**29.3**
MBRM	50.2	46.5	16.9	21.3
JEC	52.4	47.3	17.1	20.9
TagProp	69.7	58.4	20.6	28.5
FastTag	71.5	63.8	23.3	32.2
OGL+FastTag	**73.8**	**65.1**	**24.4**	**33.9**

5.3. Results

Fig. 3 shows the comparisons for the four datasets and Table 1 depicts the MAP for the different methods. We also show some qualitative results in Fig. 4. From these figures, we have the following observations:

- By learning an optimal graph from partial tags and multiple features, the performance for image and video annotation is improved significantly compared with other graph construction methods. This indicates that using partial tags and multiple features can result in a better graph.
- The improvement of OGL over other methods is more significant for IXMAS dataset than for the other datasets. This is probably because out-of-sample extension will reduce the performance gaps of different methods.
- L2Graph is a strong competitor in all cases, while LGC achieves the worst performance in all datasets. This indicates that reconstruction coefficients-based graph is better than pair-wise distance-based graphs in these datasets.
- Compared with other state-of-the-art none-graph-based methods, our OGL outperforms most of the existing approaches and achieves comparable performance with the current best result. For fair comparison, the results of TagProg are reported without the metric learning step. FastTag achieves better performance than our OGL since it uses a complex iterative algorithm to learn the annotation function, while our OGL utilizes a least square regression for simplicity to annotate novel images. When we adopt the same annotation model as in FastTag, our method (OGL+FastTag) outperforms the existing methods consistently.
- From the qualitative results in Fig. 4, we can see that OGL can annotate the videos more precisely in the IXMAS dataset than LLE. This is because OGL makes use of both tags and multiple features and automatically decides their weights, while LLE considers visual features only. In this case, OGL is more robust to some inaccurate visual features (e.g., light) or the tags, but LLE is more sensitive to these changes.

6. Conclusions and Future Work

In this work, we propose learning an optimal graph from multiple cues (i.e., partial labels and multiple features) to embed the relationships among data points more precisely. Then, OGL is incorporated with semi-supervised learning model for image and video annotation. An efficient solution is proposed for OGL, and our model is further extended by addressing out-of-sample and noisy label issues. Experiments on four public datasets show that OGL can gain 12%, 5%, 2%, 3% performance improvement over state-of-the-art algorithm in terms of MAP.

In the future, we consider incorporating OGL with other graph-based learning algorithms to test the performance.

7. Acknowledgments

The work of Jingkuan Song, Yan Yan and Nicu Sebe has been partially supported by the EC project xLiMe.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled

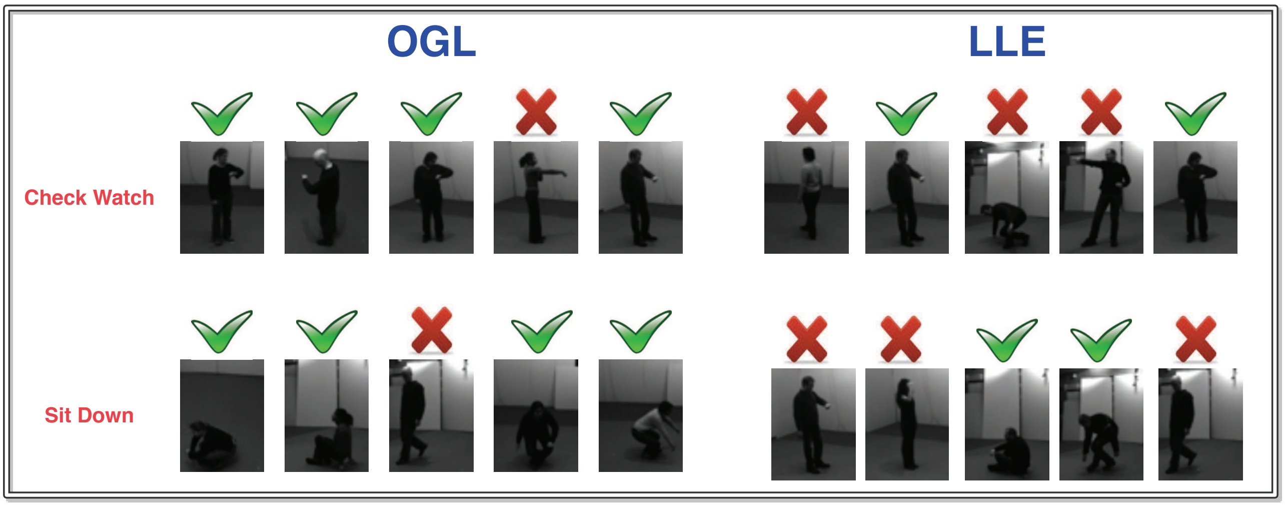


Figure 4. Qualitative results on IXMAS dataset.

and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[2] M. Chen, A. Zheng, and K. Q. Weinberger. Fast image tagging. In *ICML*, 2013.

[3] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang. Learning with l1-graph for image analysis. *IEEE TIP*, 19(4):858–866, 2010.

[4] C. Deng, R. Ji, D. Tao, X. Gao, and X. Li. Weakly supervised multi-graph learning for robust image reranking. *IEEE TMM*, 16(3):785–795, 2014.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11):2765–2781, 2013.

[6] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[7] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *CVPR*, 2007.

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

[9] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI*, 35(1):171–184, 2013.

[10] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[11] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *ICCV*, 2011.

[12] R. Liu, Z. Lin, F. de la Torre, and Z. Su. Fixed-rank representation for unsupervised visual learning. In *CVPR*, 2012.

[13] W. Liu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[14] W. Liu, J. Wang, S. Kumar, and S. Chang. Hashing with graphs. In *ICML*, 2011.

[15] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *IJCV*, 90(1):88–105, 2010.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[17] F. Nie, D. Xu, I. W.-H. Tsang, and C. Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE TIP*, 19(7):1921–1932, 2010.

[18] X. Peng, L. Zhang, and Z. Yi. Constructing l2-graph for subspace learning and segmentation. *CoRR*, abs/1209.0841, 2012.

[19] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE TPAMI*, 32(10):1832–1845, 2010.

[20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, 2005.

[22] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, 2011.

[23] J. Song, Y. Yang, X. Li, Z. Huang, and Y. Yang. Robust hashing with local models for approximate similarity search. *IEEE Trans. Cybernetics*, 44(7):1225–1236, 2014.

[24] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, 2013.

[25] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain. Image annotation by *k*nn-sparse graph-based label propagation over noisily tagged web images. *ACM TIST*, 2(2):14, 2011.

[26] R. Vidal. Subspace clustering. *IEEE Signal Process. Mag.*, 28(2):52–68, 2011.

[27] F. Wang and C. Zhang. Label propagation through linear neighborhoods. *IEEE TKDE*, 20(1):55–67, 2008.

[28] J. Wang, T. Jebara, and S.-F. Chang. Graph transduction via alternating minimization. In *ICML*, 2008.

[29] S. Yan and H. Wang. Semi-supervised learning by sparse representation. In *International Conference on Data Mining*, 2009.

[30] Y. Yan, E. Ricci, S. Ramanathan, O. Lanz, and N. Sebe. No matter where you are: Flexible graph-guided multi-task learning for multi-view head pose classification under target motion. In *ICCV*, pages 1177–1184, 2013.

[31] Y. Yan, E. Ricci, R. Subramanian, G. Liu, and N. Sebe. Multitask linear discriminant analysis for view invariant action recognition. *IEEE TIP*, 23(12):5599–5611, 2014.

[32] Y. Yang, Z. Wang, J. Yang, J. Wang, S. Chang, and T. S. Huang. Data clustering by laplacian regularized l1-graph. In *AAAI*, 2014.

[33] Y. Zhang and D.-Y. Yeung. Semisupervised generalized discriminant analysis. *IEEE TNN*, 22(8):1207–1217, 2011.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[35] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.