Yun Fu
Editor

Low-Rank and Sparse Modeling for Visual Analysis

Springer

Editor
Yun Fu
Northeastern University
Boston, MA
USA

ISBN 978-3-319-11999-1 ISBN 978-3-319-12000-3 (eBook)
DOI 10.1007/978-3-319-12000-3

Library of Congress Control Number: 2014951660

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Low-Rank Online Metric Learning

Yang Cong, Ji Liu, Junsong Yuan and Jiebo Luo

Abstract Image classification is a key problem in computer vision community. Most of the conventional visual recognition systems usually train an image classifier in an offline batch mode with all training data provided in advance. Unfortunately in many practical applications, usually only a small amount of training samples are available in the initialization stage and many more would come sequentially during the online process. Because the image data characteristics could dramatically change over time, it is important for the classifier to adapt to the new data incrementally. In this chapter, we present an online metric learning model to address the online image classification/scene recognition problem via adaptive similarity measurement. Given a number of labeled samples followed by a sequential input of unseen testing samples, the similarity metric is learned to maximize the margin of the distance among different classes of samples. By considering the low-rank constraint, our online metric learning model not only provides competitive performance compared with the state-of-the-art methods, but also guarantees to converge. A bi-linear graph is also applied to model the pair-wise similarity, and an unseen sample is labeled depending on the

Y. Cong (✉)
State Key Laboratory of Robotics, Shenyang Institute of Automation,
Chinese Academy of Sciences, Shenyang 110016, China
e-mail: congyang81@gmail.com

Y. Cong
Department of Computer Science, University of Rochester, Rochester, USA

J. Liu
Department of Computer Science, University of Rochester, Rochester 14627, USA
e-mail: jliu@cs.rochester.edu

J. Yuan
School of EEE, Nanyang Technological University, Singapore 639798, Singapore
e-mail: jsyuan@ntu.edu.sg

J. Luo
Department of Computer Science, University of Rochester, Rochester 14627, USA
e-mail: jiebo.luo@gmail.com

Y. Fu (ed.), *Low-Rank and Sparse Modeling for Visual Analysis*,
DOI 10.1007/978-3-319-12000-3_10

graph-based label propagation, while the model can also self-update using the new samples that are more confident labeled. With the ability of online learning, our methodology can well handle the large-scale streaming video data with the ability of incremental self-update. We also demonstrate that the low-rank property widely exists in natural data. In the experiments, we evaluate our model to online scene categorization and experiments on various benchmark datasets and comparisons with state-of-the-art methods demonstrate the effectiveness and efficiency of our algorithm.

Keywords Low-rank · Online learning · Metric learning · Image categorization

1 Introduction

Nowadays, machine learning technologies have been demonstrated to play a crucial role in many practical visual system. Given the training data, most of the state-of-the-art machine learning models are usually trained offline in a batch mode, which cannot be updated during the online procedure, e.g., the computer vision system for scene categorization and image classification [1–7]. Unfortunately, most of the practical systems are indeed an online system, where the property of new incoming data may deviate from the initial training data. This makes the performance of the machine learning model deteriorated over time accordingly. In order to handle such a issue, a traditional method is to re-train the machine learning model again using both existing training data and new incoming data. Obviously, this is a time-consuming way. Moreover, if the size of the training dataset is too large, it is difficult for the batch training model to handle all the data in one iteration.

To overcome these problems, online models that learn from one or a group of instances each time [8–13] provide an efficient alternative to offline re-training by incrementally updating the classifier upon the new arrivals and establishing a decision boundary that adapts to the ever-changing data. In this chapter, we focus on an adaptive similarity learner by representing the model in a matrix form, similar to metric learning, collaborative filtering, and multi-task learning. The intention of the online metric learning model is to learn a Positive Semi-definite (PSD) matrix $W \in \mathbb{R}^{d\times d}$, such that $p_1^T W p_2 \geq p_1^T W p_3$ for all $p_1, p_2, p_3 \in \mathbb{R}^d$; if p_1, p_2 are more similar and p_1, p_3 are less similar. For classification, p_1, p_2 should be from the same class and p_3 is from a different one. Essentially, the supervised online metric learner is designed to distinguish feature points with max margin as well. If all data with dimension d lie in a low dimension subspace r ($r < d$), the metric matrix with the rank less than r can distinguish any two samples if the data is distinguishable. Ideally, for data without any noise, many metric matrices with rank larger than r can distinguish it. However, training data always contains noise in practice, thus the metric matrix with a high rank would cause over-fitting and is sensitive to the noise and therefore not robust.

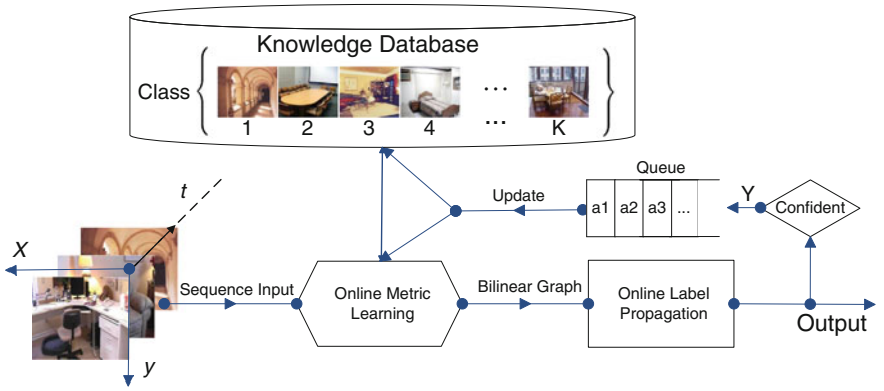

Fig. 1 A demonstration of our online learning flowchart: We first collect labeled data and train an initial model. Next, with video data arriving sequentially, we extract the feature from each image/frame and use online metric learning and label propagation to make a prediction. Finally, the samples with sufficient confident scores are inserted into the training set queue for online updating the model incrementally

It is well known that the low-rank property is often satisfied in practical data. We thus consider the low-rank constraint in our metric learning model and learn a low dimensional representation of the data in a discriminative way, where low-rank matrix models can therefore scale to handle substantially many more features and classes than with full-rank dense matrices. For classification based on our online metric learning model, we define a bi-linear graph model to predict the label of a new incoming testing sample and fuse the information of both labeled and unlabeled data in the fashion of semi-supervised learning. Then a unified framework is designed to online self-update the models, which are used to handle online scene categorization, as shown in Fig. 1. This chapter is an extension work of our previous work [14]. The main contributions of our chapter are as follows:

i. By considering the low-rank property of the data distribution, we propose a novel online metric learning model with the low-rank constraint to overcome over-fitting.
ii. We define a bi-linear graph to measure the similarity of pair-wise nodes. Different from traditional similarity graphs, such as full graph, $k-$NN and $\epsilon-$NN graphs, our bi-linear graph can maintain its accuracy for label propagation without tuning any parameters.
iii. We propose a general framework for online self-supervised learning by combining online metric learning with semi-supervised label propagation. In comparison to supervised learning with batch training, our algorithm can self-update the model incrementally and incorporate useful information from both labeled and unlabeled samples.

The organization of the rest of our chapter is as follows. In Sect. 2, we review the related work. In Sect. 3, we demonstrate that the low-rank property widely exists

in natural data. In Sect. 4, we propose our online metric learning model. We then describe the general framework of our online learning framework in Sect. 5 including online label propagation model and model updating, respectively. Section 6 reports our experimental results and comparisons with state-of-the-art methods. Finally, we summary this chapter in Sect. 7 and propose the acknowledge in Acknowledgments section.

2 Related Work

Image classification or scene categorization has been a long term research topic in both the computer vision and image processing community. Classifying scenes into categories, such as kitchen, office, coast and forests, is a challenging task due to the scale, illumination, content variations of the scenes and the ambiguities among similar types of scenes. For scene categorizations, there are mainly two key issues: image representation and similarity measurement.

For **image representation**, there are many scene descriptors. For example, various Histogram-based features have been widely adopted for image classification, such as [15, 16]. Recently, as the SIFT [17] feature are popularized in the computer vision community, most of researchers select SIFT for image representation due to the SIFT feature is invariant to scale and robust to orientation changes. There are also some multi-channel descriptors varied from traditional gray-level SIFT, such as CSIFT [18], HSV-SIFT [19], MSIFT [20] and HueSIFT [21], which is usually extract a SIFT vector from each channel, and then concatenate these vectors into a high-level multi-channel descriptor directly. The SIFT descriptor has been shown to overcome many other low-level features, such as edge feature, raw pixel intensities feature for scenes and places recognizing. For global representation of the whole image, some Bag of Words (BOW) methods are employed to postprocess the SIFT features. The spatial pyramid matching [22] are also employed to systematically incorporate spatial information where features are quantized in M discrete types using K-means clustering with M centroid by assuming to match only the same type of features. There are also kernel codebook [23, 24]. Moreover, Oliva and Torralba propose the Gist descriptor to represent the spatial structures by computing the spectral information in an image through Discrete Fourier Transform (DFT). Then the Karhunen-Loeve Transform (KLT) is used to compress the spectral signals. The Gist has been validated to achieve a good performance in recognizing outdoor scenes, e.g., coast, mountain and grass; however, its performances worse for indoor environments. To overcome this, Wu et al. [25, 26] propose the CENTRIST (CENsus TRansform hISTogram) feature, a visual descriptor that is suitable for recognizing topological places and scenes categorizes. There are also some extension works, such as mCENTRIST [27] and the combination of CENTRIST and color cues [28]. Designing an effective scene representation is beyond the scope of this chapter and we chose to adopt CENTRIST here.

For **similarity measurement**, most traditional methods for scene recognition focus on supervised learning with batch training, such as [24, 29–36], which cannot handle online processing, and would break down if the size of dataset is too large. Online algorithms have received much attention in the last decade, as they learn from one instance or sample at a time. For online supervised learning methods, Cauwenberghs et al. [11] propose a solution to the problem of training Support Vector Machines (SVMs) with a large amount of data; Utgoff et al. [37] introduce incremental decision tree classifiers that can be updated and retrained using new unseen data instances. Several methods have been proposed to extend the popular AdaBoost algorithm to the online scenario, for example complex background and appearance models [38], and visual tracking [8, 39, 40]. Moreover, there are also many practical industrial applications using online learning, e.g., [41] designs online image classifiers to handle CD imprint inspection in industrial surface inspection; [42] presents an online machine vision system for anomaly detection in sheet-metal forming processes; [43] models user preferences using online learning and also [10] combines supervised and semi-supervised online boosting trees. Learning a measurement of similarity between pairs of objects is a fundamental problem in machine learning. A large margin nearest neighbor method (LMNN) [44] is proposed to learn a Mahalanobis distance to have the k-nearest neighbors of a given sample belong to the same class while separating different-class samples by a large margin. LEGO [45], Online learning of a Mahalanobis distance using a Log-Det regularization per instance loss, is guaranteed to yield a positive semidefinite matrix. In [46], a metric learning by collapsing classes (MCML) is designed to learn a Mahalanobis distance such that same-class samples are mapped to the same point, formulated as a convex problem. Chechik et al. [9, 47, 48] design an Online Algorithm for Scalable Image Similarity learning (OASIS), for learning pairwise similarity that is fast and scales linearly with the number of objects and the number of non-zero features. However OASIS may suffer from over-fitting. Therefore, we employ the low-rank constraint to overcome overfitting accordingly and define a new online metric learning model [14].

3 The Low-Rank Property in Natural Data

In this section, we illustrate whether the low-rank property widely exists or not in real visual data. In order to calculate the video rank, we first extract 1302-d CENTRIST feature [26] with the spatial-pyramid structure from each image (or frame) and collect a $\mathbb{R}^{n\times d}$ feature matrix from each video accordingly, where n is the number of frames and $d = 1302$ in our case. Next, we use SVD decomposition to compute the eigenvalues and sort the eigenvalues in a descending order. We define the number of used eigenvalues with the accumulate eigenvalues up to 95 % of the whole as a criterion of the video rank, i.e., the less the eigenvalue is used, the lower the rank of the video is. For the video dataset, we use the Visual Place Categorization (VPC) 09 video dataset including 12 different scenarios. Following the experiment setting

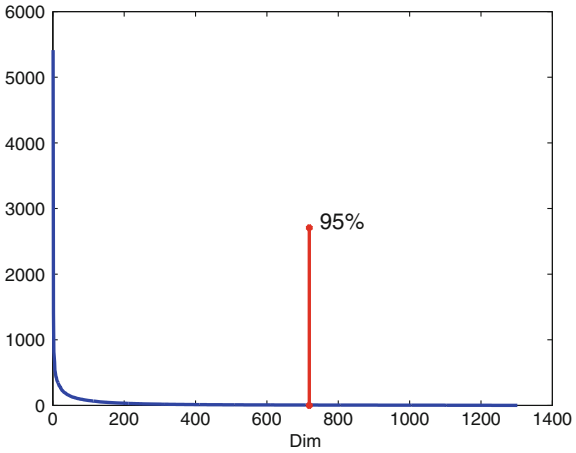

Fig. 2 An example, where the *horizontal* and *vertical* axes are the rank index and rank value, respectively. The *red bar* indicates the video rank, i.e., the index of the cumulative eigenvalue up to 95 %

Table 1 The demonstration of the low-rank property in real video data

VideoID	Full dim	Frame num	Video rank
1	1302	5888	750
2	1302	7185	740
3	1302	4546	719
4	1302	5789	736
5	1302	4166	707
6	1302	5520	723

by [25], we also adopt 5 categories for comparison in our chapter, i.e., bedroom, bathroom, kitchen, living-room and dining-room. An illustration using video1 is shown in Fig. 2, where the horizontal and vertical axes are the indexes and value of the rank, respectively. The red bar indicates the index of the cumulative sum value/energy of rank up to 95 %. The statistic results are shown in Table 1, where we totally have 6 videos and each video contains about 4 ∼7 k images. The last column of Table 1 is the video rank, i.e., the used eigenvalue number with the accumulative eigenvalues up to 95 % of the full video. Moreover, the statistic figure is shown in Fig. 3, where the horizonal axis indicates the rank of each video; the indexes "1" to "6" in the vertical axis are the video indexes corresponding to Table 1 (the last red bar "Full" means the full rank should be 1302). We can see that the rank of all the videos are all about 700 in comparison with the full rank of 1302. Therefore, we can conclude that the low-rank property should widely exist in natural visual data.

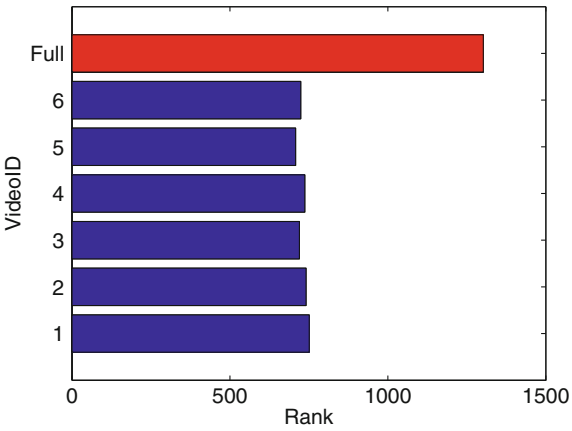

Fig. 3 The demonstration of the low-rank property in real video data. The *vertical* axis is the video id (the last *red bar* "Full" is used for comparison to show that the full rank should be 1302), and the *horizontal* axis indicates the rank. We can see that the video ranks in all cases are about 700, which is nearly the half of full rank

Table 2 Notations

ID	Definition
1	$A \in \mathbb{R}^{d \times d}$ is a symmetric matrix
2	The eigenvalue decomposition $A = U \Lambda U^T$
3	$U^T U = I$, $U \in \mathbb{R}^{d \times d}$
4	$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$ is a diagonal matrix
5	The truncate operation $\mathcal{T}_\tau(A) = U \mathcal{T}_\tau(\Lambda) U^T$
5	$(z)^+$ or $z^+ = max(0, z)$
6	$\Lambda^+ = diag(\lambda_1^+, \lambda_2^+, \ldots, \lambda_d^+)$
7	$\mathcal{D}_\tau(\Lambda) = diag((\lambda_1 - \tau)^+, (\lambda_2 - \tau)^+, \ldots, (\lambda_d - \tau)^+)$
8	$A^+ = U \Lambda^+ U^T$ and $A^- = A - A^+$
9	The shrinkage operation of A is $\mathcal{D}_\tau(A) = U \mathcal{D}_\tau(\Lambda) U^T$
10	$\mathcal{T}_\tau(\Lambda) = diag(min(\lambda_1, \tau), min(\lambda_2, \tau), \ldots, min(\lambda_d, \tau))$
11	$A = \mathcal{T}_\tau(A) + \mathcal{D}_\tau(A)$

4 Our Online Metric Learning Model

In this section, we propose an online metric learning model via low-rank constraint. We first define some notations (Table 2).

4.1 Online Metric Learning with Low-Rank Constraint (OMLLR)

The goal of Online Metric Learning (OML) is to learn a similarity function $s_W(p_i, p_j)$ parameterized by matrix W for similarity measurement, which is a bi-linear form [9, 47] as:

$$s_W(p_i, p_j) \equiv p_i^T W p_j, \tag{1}$$

where $p_i, p_j \in \mathbb{R}^d$ are the feature vectors and $W \in \mathbb{R}^{d\times d}$. s_W assigns higher scores to more similar pairs of feature vectors and vice versa. For robustness, a soft margin is given as

$$s_W(p_i, \hat{p}_i) > s_W(p_i, \bar{p}_i) + 1, \ \forall p_i, \hat{p}_i, \bar{p}_i \in P. \tag{2}$$

Here $\hat{p}_i \in P$ is more similar to $p_i \in P$ than $\bar{p}_i \in P$. In our case, p_i and $\hat{p}_i$ belong to the same class; while $p_i, \bar{p}_i$ are from different classes. The hinge loss function $l_W(\cdot, \cdot, \cdot)$ is used to measure the cost:

$$l_W(p_i, \hat{p}_i, \bar{p}_i) = \max\big(0, 1 - s_W(p_i, \hat{p}_i) + s_W(p_i, \bar{p}_i)\big). \tag{3}$$

For the Online Algorithm for Scalable Image Similarity learning (OASIS) in [9, 47], the Passive-Aggressive algorithm is used to minimize the global loss l_W. The OASIS model solves the following convex problem with a soft margin:

$$\begin{aligned} W^t &= \arg\min_W \frac{1}{2}\|W - W^{t-1}\|_F^2 + \mu\xi, \\ &\text{s.t. } l_W(p_t, \hat{p}_t, \bar{p}_t) \le \xi \text{ and } \xi \ge 0 \end{aligned} \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm ($\|W\|_F = \sqrt{\sum_{i,j} W_{ij}^2}$) and μ is the tuning parameter. To minimize the global loss l_W in Eq. (4), the Passive-Aggressive algorithm is utilized:

$$\begin{cases} W = W^{t-1} + \tau V_t \\ \tau = \min\left\{\mu, \frac{l_{W^{t-1}}(p_t, p_t^+, p_t^-)}{\|V_t\|^2}\right\}. \end{cases} \tag{5}$$

In the initialization, W is set to an identity matrix $W^0 = I_{d\times d}$. Next, the OASIS model randomly samples the triplet $(p_t, \hat{p}_t, \bar{p}_t)$ iteratively for online learning. OASIS is efficient for optimization with computational complexity of $O(n^2)$, but it has two main drawbacks: (1) its performance may fluctuate with each iteration because the sampled triplet $(p_t, \hat{p}_t, \bar{p}_t)$ cannot guarantee the effectiveness; (2) the model itself may suffer from overfitting, because W could have redundant degrees of freedom especially when the object templates lie in a low dimensional subspace of $\mathbb{R}^d$. Theorem 1 of our previous work has shown that for the data in a subspace with dimension $r < d$, a metric matrix with a rank at most r can determine the similarity measurement.

Theorem 1 *For any matrix $X \in \mathbb{R}^{n\times d}$ with rank r and any Positive Semi-definite (PSD) matrix $W \in \mathbb{R}^{d\times d}$, there exists a PSD matrix $Q \in \mathbb{R}^{d\times d}$ with rank$(Q) \le r$ such that*

$$X^T W X = X^T Q X.$$

In practice, each column of X is a data point $p_t \in \mathbb{R}^d$, and we have $X_i^T W X_j = X_i^T Q X_j$. It means that for pair-wise similarity measurement of X_i and X_j, the metric matrix W is not unique.

If we construct the data matrix X from $\{p_t, \hat{p}_t, \bar{p}_t \mid \text{all } t\}$ (each column of X is a data point $p_t \in \mathbb{R}^d$) with a metric W, then we can always find a metric Q whose rank is at most r such that $X_i^T W X_j = X_i^T Q X_j$.

Consider the training data with K classes $P_1, \ldots, P_K$ and let $P = \cup_{i=1}^K P_i$. Define the hinge loss function as $l(W, t) = max(0, 1 - p_t^T W \hat{p}_t + p_t^T W \bar{p}_t)$ like Eq. (3) where $\hat{p}_t, \bar{p}_t, p_t \in \mathbb{R}^d$, $W \in \mathbb{R}^{d \times d}$, and t is a random index, which is usually sampled uniformly from a index set T that includes K classes.

In order to estimate the metric matrix with a low-rank property, a natural idea is to solve the following optimization problem:

$$\begin{aligned} \min_W &: f(W) := \mathbb{E}_t[l(W, t)] + \gamma rank(W) \\ s.t. &: W \succeq 0. \end{aligned} \tag{6}$$

Unfortunately, the optimization problem in Eq. (6) is non-convex and NP-hard. A conventional way is to use the trace norm $\| \|_*$ to approximate the rank function $rank(W)$, which makes the problem tractable:

$$\begin{aligned} \min_W &: f(W) := \mathbb{E}_t[l(W, t)] + \gamma \|W\|_* \\ s.t. &: W \succeq 0. \end{aligned} \tag{7}$$

If t follows the uniform distribution over the index set Φ, then

$$\mathbb{E}_t[l(W, t)] = \frac{1}{|\Phi|} \sum_{t \in \Phi} l(W, t) \tag{8}$$

If one can evaluate the subdifferential of $\mathbb{E}[l(W, t)]$ at each step, then the proximal operation can be applied to solve the problem in Eq. (7):

$$\begin{aligned} W^{i+1} = arg \min_W &: \frac{1}{2} \|W - W^i + \alpha^i \partial \mathbb{E}_t[l(W^i, t)]\|^2 \\ &+ \alpha^i \gamma \|W\|_* \\ s.t. &: W \succeq 0 \end{aligned}$$

Define the proximal operation as

$$prox_{P,\Omega}(x) = arg \min_{y \in \Omega} \frac{1}{2} \|y - x\|_F^2 + P(y). \tag{9}$$

In our case, $P(W) = \alpha^i \gamma \|W\|_*$ and $\Omega = \{W \mid W \succeq 0\}$. Then we have $W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i \partial \mathbb{E}_t[l(W^i, t)])$.

The gradient of $\mathbb{E}_t[l(W^i, t)]$ is not computable sometimes, e.g., some data samples $\hat{p}_t, \bar{p}_t, p_t$ are unavailable in the ith iteration, or it is too expensive to evaluate $\partial\mathbb{E}_t[l(W^i, t)]$ due to the large-scale training data. In order to handle this issue, the stochastic algorithm uses $\partial l(W^i, t)$ to approximate $\partial\mathbb{E}_t[l(W^i, t)]$ where t is randomly generated at each iteration, due to $\partial\mathbb{E}_t[l(W^i, t)] = \mathbb{E}_t[\partial l(W^i, t)]$. Thus, in the stochastic algorithm, the basic updating rule in each iteration is

$$W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i \partial l(W^i, t)) \tag{10}$$

We can summarize the algorithm in Algorithm 1.

Algorithm 1 Online Metric Learning with low-rank

Require: $\hat{p}_t, \bar{p}_t, p_t$ are the random sampled triplet for all t
γ and α are tuning parameters
Ensure: W
1: Initialize $i = 0$ and $W^0 = I \in \mathbb{R}^{d\times d}$
2: Repeat the following steps until exceeding the maximum iteration number
3: Generate t from its distribution
4: $W^{i+1} = prox_{P(W),\Omega}\left(W^i - \alpha^i \partial l(W^i, t)\right)$
5: $i = i + 1$

Step 4 is the key step in this algorithm. First, one can verify that

$$\partial l(W, t) = \begin{cases} (\bar{p}_t - \hat{p}_t)p_t^T, & l(W, t) > 0; \\ \left[0,\ (\bar{p}_t - \hat{p}_t)p_t^T\right], & l(W, t) = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Note that $\partial l(W, t) = 0$ is a range when $l(W, t) = 0$. In this case, $\partial l(W, t)$ can take any value in this range. Theorem 2 introduces the closed form of $W^{i+1} = prox_{P,\Omega}(W^i - \alpha^i \partial l(W^i, t))$:

Theorem 2 *Let* $P(W) = \|W\|_*$ *and* $\Omega = \{W \mid W \succeq 0\}$. *We have*

$$prox_{\gamma P,\Omega}(C) = \mathcal{D}_\gamma\left(\frac{1}{2}(C + C^T)\right). \tag{12}$$

Another remaining issue is how to choose the step size α^i. One of the conventional way is to let $\alpha^i = \Omega(1/\sqrt{i})$, which can lead to the optimal convergence rate as $\mathbb{E}[f(\overline{W}) - f(W^*)] \leq O\left(\frac{1}{\sqrt{|\Phi|}}\right)$, where W^* is the optimal solution and

$$\overline{W} = \frac{1}{|\Phi|}\sum_{i=1}^{\Phi} W_i \tag{13}$$

5 The Flowchart of Our Algorithm

In this section, we will introduce the flowchart of our online image classification algorithm as shown in Fig. 4, which uses metric learning to measure the similarity and adopts semi-supervised learning to label the testing samples. Our algorithm mainly includes two phases: batch initial training phase and online prediction phase. During the batch initial training, each image is assigned a label and useful features are extracted and stored as feature vectors along with their labels. We then perform batch training to obtain an initial metric learner with the low-rank constraint, i.e., the matrix W for similarity measurement. During the online training phase, features are also extracted from each sequentially incoming image, and depending on whether the data has a label or not, the proposed supervised and semi-supervised classifiers will be used to self-update the metric learner W. For an unlabeled sample, we measure the similarity between it and each of the initial training samples and propagate the label using our bi-linear graph accordingly. Next, those samples with high confidence scores are also used to update W. All the labeled samples are used for updating, where the updating procedure is similar to the batch initial training. Such a process iterates during online processing. The online learning phase will stop if the prediction performance reaches a desired level. Generally, there are mainly two key technical issues, online metric learning and label propagation, which are discussed below.

5.1 Online Label Propagation

Depending on the similarity measured by OMLLR above, we adopt the graph-based semi-supervised learning (also called label propagation) to make a more accurate

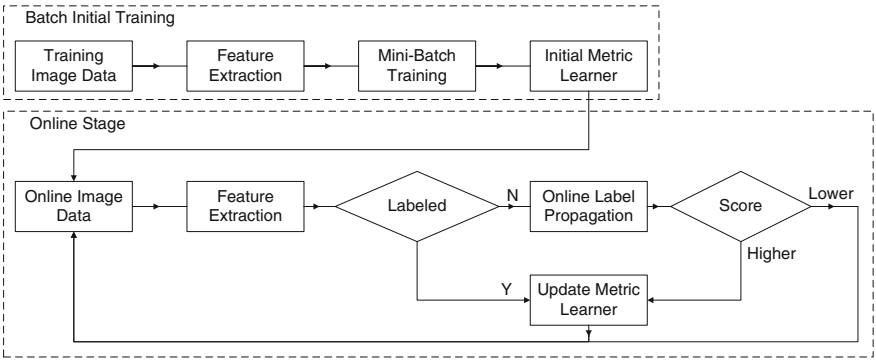

Fig. 4 The work flow of the proposed online learning algorithm. The min-batch training is used for initialized training procedure and for the online testing stage, the sample with sufficient confident score is selected to online update the model itself

prediction, which associates the information of both the labeled data and unlabeled data. For similarity graph, we define a new bi-linear graph using OMLLR:

Definition 1 Bi-linear Graph: Assume the similarity of pairwise points $\forall\ i,\ j,\ \ 1 \leq i, j \leq n,\ i \neq j$ is defined as

$$S_{i,j} \ = \ \max(0,\ S_w(i,j)) \ = \ \max(0,\ p_i^T W p_j). \tag{14}$$

For $p_i \in P,\ i \in [1, \ldots, N]$, we obtain a matrix $\{S_{ij}, 1 \leq i, j \leq N\}$, where its symmetric version is $S_{i,j} = (S_{i,j} + S_{j,i})/2$.

In comparison with other traditional graph models, e.g., $k-$NN or $\epsilon-$NN graph, which are either sensitive to tuning parameters (e.g., σ) or instable to define a suitable graph structure without enough prior knowledge (e.g., k or ϵ), our bi-linear graph can maintain the accuracy without tuning parameters or prior knowledge of the topology graph.

To predict the label of the new data, we define G $=$ (V, E), where V denotes $n = n_l + n_u$ feature vectors (n_l labeled and n_u unlabeled); and E contains the edges of every pair of nodes measuring the pairwise similarity. Suppose we have $\Psi = \{1, 2, \ldots, \mathrm{K}\}$ classes. Let $F = \binom{F_l}{F_u} \in \mathbb{R}^{(n_l+n_u)\times K}$, where $F_l = [f_1, f_2, \ldots, f_{n_l}]^T \in \mathbb{R}^{n_l \times K}$ denotes the label matrix of the labeled data, and $F_u = [f_1, f_2, \ldots, f_{n_u}]^T \in \mathbb{R}^{n_u \times K}$ is the label matrix of unlabeled data needed to be predicted. In order to facilitate the calculation, we first normalize the similarity matrix S as,

$$P_{ij} = P(i \to j) = \frac{S_{ij}}{\sum_{k=1}^{n} S_{ik}}. \tag{15}$$

The matrix $P \in \mathbb{R}^{n\times n}$ can be split into labeled and unlabeled sub-matrices,

$$P = \begin{bmatrix} P_{ll} & P_{lu} \\ P_{ul} & P_{uu} \end{bmatrix}, \tag{16}$$

where $P_{ll} \in \mathbb{R}^{n_l \times n_l}$, $P_{lu} \in \mathbb{R}^{n_l \times n_u}$, $P_{ul} \in \mathbb{R}^{n_u \times n_l}$ and $P_{uu} \in \mathbb{R}^{n_u \times n_u}$. For label propagation, we have

$$F_u^{t+1} \leftarrow P_{uu} F_u^t + P_{ul} F_l. \tag{17}$$

When t approaches infinity, we have

$$F_u = \lim_{t\to\infty} (P_{uu})^t F_u^0 + (\sum_{i=1}^{t} P_{uu}^{(i-1)}) P_{ul} F_l, \tag{18}$$

where F_u^0 is the initial value of F_u. Since P is a row normalized matrix, the sum of each row of the sub-matrix $(P_{uu})^n$ approaches to zero. As a result, the first item of Eq. (18) converges to zero, $(P_{uu})^n F_u^0 \to 0$. Furthermore, the second item of Eq. (17)

Algorithm 2 Testing & Online Learning

Require: Query sample q, similar matrix W, training set $\{p_i\}$, threshold T_ξ
Ensure: $W^\star$
1: Generate Bi-linear Graph S by Eq. (14)
2: $c_q^* = \arg\max E_c(q)$ by Eq. (21)
3: **if** $E(c_q)/E(\overline{c}_q) > T_\xi$ **then**
4: Insert $q \Rightarrow$ queue Q
5: **end if**
6: **if** Full (Q) **then**
7: Update (Q)
8: Insert $Q \Rightarrow \{p_i\}$ and clear Q
9: **end if**
10: **return** $W^\star = W$
11: Function Update (Q)
12: Set $i = 1$
13: **while** $i <$ ITER-MAX $\bigcap$ $\|W^i - W^{i-1}\|_F < T_w$ **do**
14: Get sample $q_i \in Q$, $q_i^+ \in c_{q_i}$ and $q_i^- \in \overline{c}_{q_i}$
15: Update W by Algorithm 1
16: $i = i + 1$
17: **end while**

can be written as

$$F_u = (I - P_{uu})^{-1} P_{ul} F_l.$$

For online predicting the label of the sequentially input sample, we have $n_u = 1$, thus $P_{uu} \in \mathbb{R}^{1\times 1}$ is a fixed real number and $(I - P_{uu})^{-1}$ is a constant if P_{uu} is not equal to 1, so

$$F_u \propto P_{ul} F_l. \tag{19}$$

Equation (19) is also consistent with the energy function we defined:

$$\mathrm{E}_c(x_i) = \sum_{j=1}^{n} \delta_c(j) S_{i,j}, \ \ \delta_c(i) = \begin{cases} 1, & i \in c \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

where $c \in \{1, \ldots, \mathrm{K}\}$; x_i denotes the query sample; $S_{x,j}, j = \{1, \ldots, n\}$ is the bi-linear graph; and $\delta_c(i)$ is an indicate function. $E_c(x)$ is the energy function, which measures the cost of x belonging to class c. Thus, given x, the optimal solution $c^\star$ is the one maximizing the energy $E_c(x)$, as

$$c_x^\star = \arg\max_c E_c(x). \tag{21}$$

5.2 Updating

In this section, we describe the updating scheme, which includes both the online metric learning model updating and the knowledge database updating as shown in Fig. 4 and Algorithm 2. Depending on the property of each testing sample, each testing sample can be classified into labeled or unlabeled sample, where all the labeled testing samples are used to update the model and the knowledge database; and for the unlabeled testing sample, it will be used to update the model if it satisfies the following criterion:

$$E_{c^\star}(q) > T_\xi \times E_{\overline{c}}(q), \quad \forall \overline{c},\ \overline{c} \neq c^\star. \tag{22}$$

Equation (22) means that the score of q^* from some category is sufficiently higher than that from other categories ($T_\xi = 1.2$ in this chapter). All samples used to update are pushed into a queue Q following a first-in first-out policy. When the queue Q is full, the matrix W of the model will be iteratively updated using both the labeled data and unlabeled data samples with high confidence scores together, and all the samples in Q will be added into the knowledge database for backup. Generally, this self-supervised online updating scheme is processed frame-by-frame, and by tuning the length L of the Q, we can select to update the online model incrementally ($L = 1$) or with mini-batch training ($L > 1$).

6 Experiments

In order to validate the effectiveness of our proposed approach, we employ several experiments and comparisons in this section, where all experiments are depending on three types of dataset as follows:

- Synthesized data: for a fair comparison, we first randomly generated synthesized data.
- Scene categorization dataset: in order to evaluate the effectiveness of our approach for scene categorization, we select both the image dataset (i.e., the 8-class sports image dataset) and video dataset (i.e., the Visual Place Categorization (VPC) 09 video dataset, which is captured in the same fashion as a real online system). From each image/frame, we extract a global CENTRIST feature [26] for image representation. The CENTRIST is in total 1302-d with the spatial-pyramid structure and we only adopt the first level of 42-d in this chapter.
- Image classification dataset: Our proposed online approach can also be used for image classification, where we choice the popular image classification dataset, i.e., Caltech 256 dataset. For image representation, we adopt the same feature used in [48] for a fair comparison, which is a spare representation based on the framework of local descriptors by combing the color histogram and texture histogram with the feature dimension as 1000.

6.1 Evaluation Criterion

We compare our method Online Metric Learning via low-rank (OMLLR) with the state-of-the-art methods including both online learning methods and batch training methods. The accuracy is defined by Eq. (23):

$$Acc = \frac{\sharp\{\text{correct categorizations samples}\}}{\sharp\{\text{total number of samples}\}} \tag{23}$$

where $\sharp\{\}$ denotes the number of samples. For batch training methods, we can generate only one final result of accuracy as defined in Eq. (23) depending on the model trained in the initialization; and for online learning methods, as the model is incrementally updated step-by-step, the performance of the online model will fluctuate with the iterations and generally the overall trend is getting better. Therefore, we adopt the model with the highest accuracy for comparison.

$$\begin{aligned} &\text{OASIS:} \quad W_{OASIS} = \arg\max_{W_j} Acc(j),\ j \in \{1, \ldots, N\} \\ &\text{LMNN:} \quad W_{LMNN} = \arg\max_{W_j} Acc(j),\ j \in \{1, \ldots, N\} \end{aligned} \tag{24}$$

where j is the index of iteration from 1 to N, and W_j is the matrix generated in each iteration.

For our OMLLR, because of our intention is to overcome the model fluctuation each iteration by improving the expectation of the model with the highest accuracy, we adopt two criteria for comparison as in Eq. (25):

$$\begin{aligned} &\text{Ours1:} \quad \overline{W} = \frac{\sum_{i=1}^{N} \alpha_i W_i}{\sum_{i=1}^{N} \alpha_i} \\ &\text{Ours2:} \quad W_{max} = \arg\max_{W_i} Acc(i),\ i \in \{1, \ldots, N\} \end{aligned} \tag{25}$$

where i is the iteration index from 1 to N, $\alpha_i = 1\sqrt{i}$ and W_i is the matrix generated by each iteration. The weighted W, "Ours1", is the expectation of the model W, which guarantees to convergence in theory; and "Ours2" is the same as Eq. (24), i.e., the model W with the highest accuracy Acc.

6.2 Synthesized Data

At first, we utilize use the synthesized data to evaluate the performance of our online metric learning model, OMLLR. We generate the synthesized data including two classes, i.e., positive and negative. In order to generate low-rank synthesized data, all data is first sampled from a low-dimensional multivariate normal distribution with

Table 3 The results of comparisons by varying the dimension and rank of features, where Ours1 is the result of our method using weighted W; Ours2 is the result of W with the highest accuracy; OASIS is the result of classical online metric learning [9, 47]

Method	Dim = 10	20	40	60	100	200
	Rank = 5	10	20	30	50	100
Ours1	81.75	99.50	98.75	89.25	87.25	85.75
Ours2	**86.25**	**99.50**	**99.00**	**89.50**	**90.25**	**86.50**
Ours1 ($\gamma = 0$)	83.5	96.50	96.50	86.50	87.25	84.75
OASIS	82.25	97.00	96.00	85.50	85.50	85.00
K-NN	75.25	97.75	95.75	83.25	80.75	76.25

The K-NN (K nearest neighbor method) is adopted as a benchmark here

full rank r, where the mean and covariance matrices are randomly generated by a uniform distribution. We then embed them into a high dimensional feature space using random projection. The random projection is achieved by multiplying the low-rank data with a high dimensional randomly generated transformation matrix. The size of both training and testing samples is 1000, so we have totally 2000 samples. Moreover, we add some Gaussian noisy artificially into the synthesized data as well for a fair comparison.

We demonstrate the result in Table 3, where our methods "Ours1" (the weighted W) and "Ours2" (the best result of W) are compared with the other state-of-the-art methods including both the online method OASIS [9, 47] and the traditional baseline K-NN. The groundtruth of the dataset "rank" varies from 5 to 100, and the corresponding feature dimension "dim" is twice of the rank. We can see that the accuracy of "Ours1" is lower than that of the "Ours2", but outperforms the classical online learning method (OASIS) and the benchmark batch training method, K-NN. In Fig. 5, we fix the feature dimension "dim" to 200, and vary the rank from 40 to 200 (with the interval of 20). The results are similar to those in Table 3, i.e., the accuracy of the benchmark method K-NN is the worst one, and our methods both "Ours1" and "Ours2" outperform the classical OASIS. Another interesting point is that, when the feature dimension is fixed, the lower the data rank, the greater the gap of the accuracy between ours and OASIS, which justifies the effectiveness of the low-rank

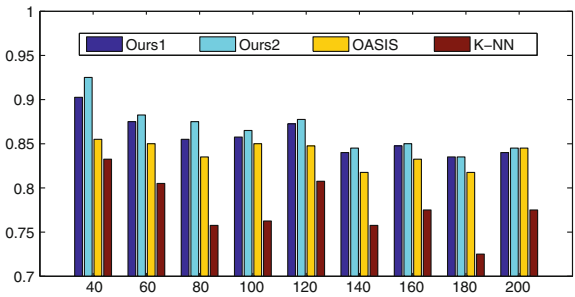

Fig. 5 Comparison of the accuracy between our methods and the state-of-the-art methods when varying the rank and fixing the feature dimension (dim = 200), where the y-axis is the accuracy and the x-axis denotes the rank

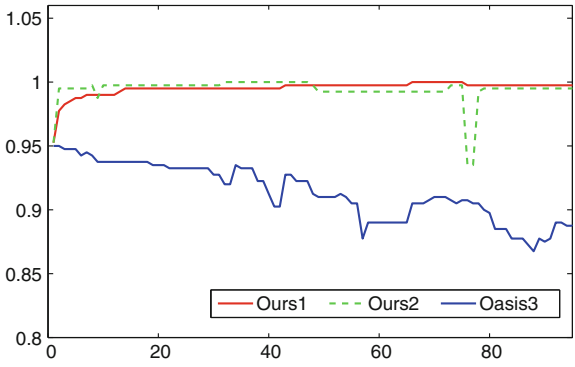

Fig. 6 An example of the simulation result, where the x-axis is the number of iterations (10 k per step) and the y-axis is the accuracy

constraint in our method. Therefore, we can conclude that our methodology can still work well for high dimensional real data having low-rank property. Figure 6 shows an example of online learning, our methodology not only outperforms OASIS, but also converges after only several iterations, e.g., "Ours1".

6.3 Sport 8 Dataset

The Sport 8 dataset [29] contains images from eight categories including badminton, bocce, croquet, polo, rock climbing, rowing, sailing, and snowboarding as shown in Fig. 7. The minimum number of category has 137 images and the maximum number of category contain 250 images. For training the initialized W in the initialization, we randomly sample 50 images from each category and leave the remaining images for testing. Figure 8 illustrates the confusion matrix, where the accuracy scores are from 64 to 91 % with an average accuracy as 75.2 %. The statistic results are shown in Table 4, where we compare our algorithm with the state-of-the-art methods and our approach outperforms other methods such as Li et al. [29], Cheng's [49] using an L1-graph based semi-supervised learning, and OASIS [9, 47]. The performance of our approach is a bit lower than Wu et al. [26], this is because Wu's [26] based on the RBF kernel uses more training samples with a high feature dimension 1302

Fig. 7 Sample images from the Sport 8 datasets, including badminton, bocce, croquet, polo, rock climbing, rowing, sailing, and snowboarding

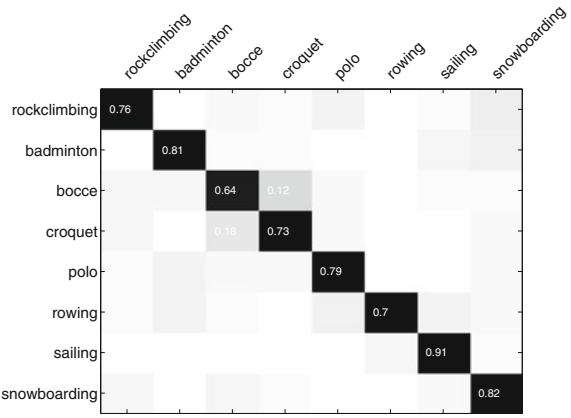

Fig. 8 Confusion matrix for the Sport 8 dataset, where the label of each row is the ground truth and the label of each column is the predicted category. The average accuracy is 77.03 %, and random chance is 12.5 %. For a better view, please check the electronic version

Table 4 The accuracy of the Sport 8 dataset

Method	Training type	Accuracy (%)
Li [29]	Batch	73.4
Wu [26]	Batch	78.2
Cheng [49]	Semi-supervised+Batch	73.2
OASIS [9, 47]	Online	69.40
Ours1	Online	75.06
Ours2	Online	77.03

for image representation, while ours uses much fewer training samples and lower feature dimension with incremental updating. An example of online learning is shown in Fig. 9, where both our methods, i.e., "Ours1" and "Ours2" outperform OASIS in every iteration and converge as well, in contrast OASIS make a little bit drift at 150

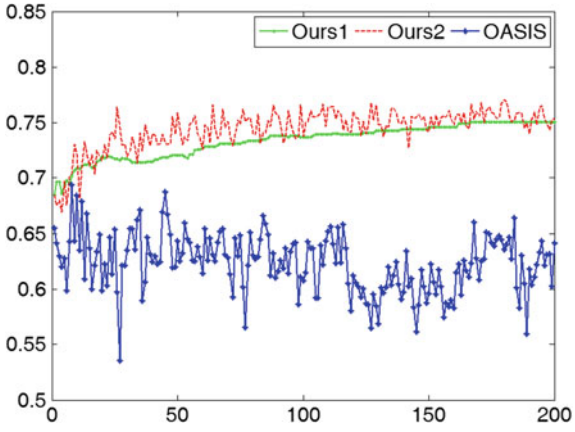

Fig. 9 The comparison of our OMLLR with OASIS using the Sport 8 dataset, where the x-axis is the number of iteration (10 k per step) and the y-axis is the accuracy

step (10 k per step). Moreover, the performance of "Ours1" nearly has no fluctuation for each iteration and still keeps up improvement the performance, this is because we intend to pursuit the model expectation with a higher accuracy and overcome the model fluctuation.

6.4 Visual Place Categorization (VPC) 09 Dataset

For the video scene dataset, we utilize the Visual Place Categorization (VPC) 09 dataset [51], which is captured using a rolling tripod plus a camera to mimic a robot. Therefore the working fashion of VPC 09 dataset is the same as an online system. The VPC dataset was collected from 6 different home, and each includes 12 different scenarios (bathroom, bedroom, closet, dining-room, exercise-room, family-room, kitchen, living-room, media-room, workspace and transition). The VPC dataset was compressed in JPEG (95 % quality) images with the resolution of each image as 1280×720.

We compare our online method (OMLLR) with the state-of-the-art methods, including two online metric learning methods i.e., OASIS [9, 47] and LMNN [50], and also some batch training methods such as K-Nearest Neighbor, 1-NN and 5-NN, and Wu's method [25]. All the experiment configuration follow by the recommendation of [25], therefore we also adopt 5 categories for comparison in our chapter, i.e., bedroom, bathroom, kitchen, living-room and dining-room. Next, a leave one out cross validation strategy is adopted to validate the performance of our algorithm, and each methods was repeated 6 times. In each run, one home was reserved for testing in turn and all other 5 homes were combined together to generate the whole training set. The overall accuracy of our online learning system is the average of the 6 individual homes.

We first compare our method with OASIS [9, 47] as shown in Fig. 10, where all the online learning models are run for 3 million iterations, and each subfigure corresponds to home 1 to home 6. In 3 million iterations of Fig. 10, the accuracy of our online model fluctuates in each iteration and the accuracy of both "Ours1" and "Ours2" outperforms OASIS in all the cases. Although the accuracy of the expectation of the model "Ours1" is not better than the best one "Ours2", it is always better than those of the other iterations, especially for Home3. Moreover, "Ours1" guarantees to converge, and the property of convergence is critical for an online algorithm in practice.

The comparisons of the accuracy for each category and each home are shown in Tables 5 and 6 respectively, where for the average accuracy of 6 categories, both "Ours1" and "Ours2" outperform other online learning methods, e.g., OASIS [9, 47], LMNN [50] and also K-NN based batch training methods (1-NN and 5-NN); and for IROS [25] using the batch training model, the accuracy of "Ours2" is better than that of IROS. In general, the accuracy of online learning models is always worse than the that of batch training methods, but the performance of our OMLLR is acceptable.

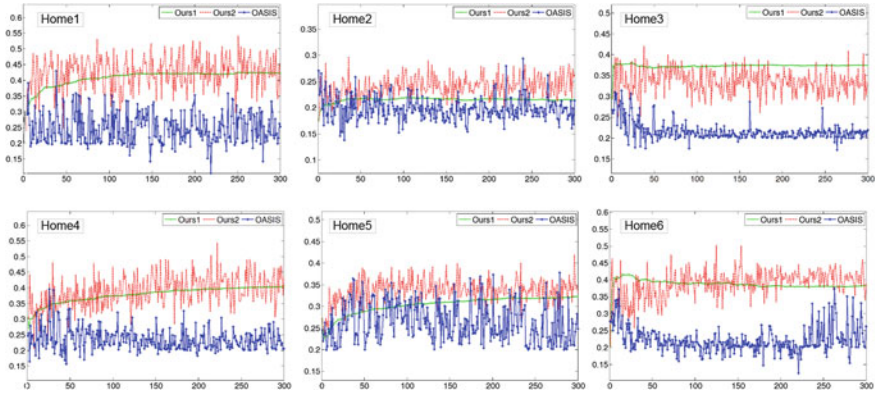

Fig. 10 The comparison of the accuracy between our OMLLR and OASIS [9, 47] for home1–6. In each figure, the x-axis corresponds to the iteration steps (10 k for each) and the y-axis is the current accuracy, where the accuracy of "Ours1", "Ours2" and OASIS is denoted by sold *green line*, *dash red line* and *dash blue line*, respectively

Table 5 The comparison of the average accuracy of our OMLLR and the state-of-the-art methods using VPC 09 dataset for each home

Filter	Train	Methods	Home1	Home2	Home3	Home4	Home5	Home6
No	Online	Ours1	42.36	21.53	37.53	40.43	32.22	38.28
		Ours2	54.50	31.12	42.89	54.99	41.95	51.13
		OASIS [9, 47]	25.33	21.32	21.99	20.57	24.84	39.18
		LMNN [50]	39.41	28.75	36.79	39.06	30.74	34.88
No	Batch	IROS [25]	44.77	33.33	40.68	43.28	41.10	48.07
		1-NN	41.83	27.48	33.96	38.66	30.85	29.70
		5-NN	41.18	28.23	34.33	39.82	31.62	31.56
Yes		Ours1	46.03	21.66	38.59	41.95	33.05	41.29
		Ours2	59.65	31.97	44.88	60.48	43.99	57.10
		IROS [25]	44.58	35.89	40.96	49.93	46.91	55.46

For the issue of frame-level scene classification, the label of consecutive frames has high correlation. In [25], Wu et al. use a temporal smooth to improve the accuracy of the coarse result, and for us, we only adopt a simple median filter for frame-level temporal smooth with filter width as 5 frames. After the operation of temporal smooth filter, the accuracy of both online learning and batch training improves, and "Ours2" is still better than IROS. Tables 7 and 8 are the specific results by "Ours1" and "Ours2". As the testing and training samples are from different scenes [25], e.g., to test Home 1, the training samples include images from Home 2 to 6. Most of the results are lower than 50 %, and both "Ours1" and "Ours2" outperform other methods. The frame-level results of scene categorization for VPC 09 are shown in Fig. 14. The images of the left column are examples of each home; and each figure

Table 6 The comparison of the average accuracy of our OMLLR and the state-of-the-art methods using VPC 09 dataset for each category

Filter	Train	Methods	Bed	Bath	Kitchen	Living	Dining	Avg
No	Online	Ours1	44.27	57.83	17.75	41.60	15.50	35.39
		Ours2	44.09	66.76	26.22	50.63	42.77	**46.09**
		OASIS [9, 47]	25.92	6.02	3.47	82.28	10.00	25.54
		LMNN [50]	41.44	51.23	26.02	38.21	17.80	34.94
No	Batch	IROS [25]	48.13	65.71	46.56	29.18	19.78	41.87
		1-NN	40.69	46.38	26.92	40.92	13.81	33.75
		5-NN	39.21	46.32	28.78	44.94	13.04	34.46
Yes		Ours1	41.12	63.04	18.52	50.06	12.74	37.10
		Ours2	43.33	72.60	31.79	58.30	42.37	**49.68**
		IROS [25]	64.89	74.77	48.24	20.59	19.61	45.62

Table 7 Categorization accuracy (Ours1) of all homes and categories when the Bayesian filtering is not used

	Bed	Bath	Kitchen	Living	Dining	Average
Home1	28.03	83.51	12.24	**95.12**	79.34	59.65
Home2	28.60	**81.15**	9.92	27.44	12.72	31.97
Home3	50.67	**89.89**	29.82	15.34	38.69	44.88
Home4	23.21	56.60	79.37	**92.78**	50.46	60.48
Home5	**81.79**	57.51	14.06	37.59	29.00	43.99
Home6	47.71	66.96	45.32	**81.53**	44.00	57.10
Average	43.33	**72.60**	31.79	58.30	42.37	49.68

Table 8 Categorization accuracy (Ours2) of all homes and categories when the Bayesian filtering is not used

	Bed	Bath	Kitchen	Living	Dining	Average
Home1	30.97	75.08	13.49	**80.49**	72.46	54.50
Home2	31.44	**71.51**	8.70	21.32	22.61	31.12
Home3	48.36	**87.93**	25.79	15.02	37.33	42.89
Home4	27.86	48.23	65.02	**85.39**	48.43	54.99
Home5	**77.91**	55.36	12.40	33.58	30.49	41.95
Home6	48.00	62.45	31.91	**67.97**	45.33	51.13
Average	44.09	**66.76**	26.22	50.63	42.77	46.09

of the right column is the frame-level result, where the x-axis is the frame index and the y-axis is the 5 class labels (bed, bath, kitchen, living and dining correspond to label 1, 2, 3, 5 and 6 with label 4 absent), and the red and blue line correspond to our

predicted result and the ground truth, respectively. So the more overlapping of red and blue lines, the higher the accuracy of our model.

6.5 Caltech 256

For the image classification, we also test our OMLLR using the Caltech 256 dataset [52], which consists of 30607 images from 257 categories and is evaluated by humans in order to ensure image quality and relevance. Following [48], we also tested on subsets of classes from Caltech 256, i.e.,

- 10 classes: bear, skyscraper, billiards, yo-yo, minotaur, roulette-wheel, hamburger, laptop-101, hummingbird, blimp.
- 20 classes: airplanes-101, mars, homer-simpson, hourglass, waterfall, helicopter-101, mountain-bike starfish-101, teapot, pyramid, refrigirator, cowboy-hat, giraffe, joy-stick, crab-101, birdbath, fighter-jet, tuning-fork, iguana, dog.
- 50 classes: car-side-101, tower-pisa, hibiscus, saturn, menorah-101, rainbow, cartman, chandelier-101, backpack, grapes, laptop-101, telephone-box, binoculars, helicopter-101, paper-shredder, eiffel-tower, top-hat, tomato, star-fish-101, hot-air-balloon, tweezer, picnic-table, elk, kangaroo-101, mattress, toaster, electric-guitar-101, bathtub, gorilla, jesus-christ, cormorant, mandolin, light-house, cake, tricycle, speed-boat, computer-mouse, superman, chimp, pram, friedegg, fighter-jet, unicorn, greyhound, grasshopper, goose, iguana, drinking-straw, snake, hotdog.

For each set, images from each class are split into a training set of 40 images and a test set of 25 images. A cross-validation procedure is also adopted to select the values of hyper parameters. For our OMLLR, the regularization parameter γ in Eq. (6) is in the set of $\gamma \in \{0.1, 0.01, 0.001, 0.001\}$.

For evaluation, a standard ranking precision measures based on nearest neighbors is also used. For each query image in the test set, all other training images are ranked according to their similarity to the query image. The number of same-class images among the top k images (the k nearest neighbors, e.g. 1, 10, 50) is computed. When averaged across test images (either within or across classes), this yields a measure known as precision-at-top-k, providing a precision curve as a function of the rank k. We also calculate the mean average precision (mAP), a widely used criterion in the information retrieval community, where the precision-at-top-k is first calculated for each test image and averaged over all positions k that have a positive sample.

Our method, OMLLR is compared with the state-of-the-art online metric learning methods, including OASIS [9, 47, 48], LMNN [44], LEGO [45], MCML [46] and Euclidean (the standard Euclidean distance in feature space). The statistic result is proposed in Table 9, where our OMLLR is the result of the expectation of the model W, i.e., "Ours1", and OMLLR($\gamma = 0$) is used for justify the efficiency of low-rank constraint, please check Sect. 6.6 for details. Our OMLLR outperforms all state-of-the-arts for the full range of k. Another interesting thing is that our performance gain is decreased with the increase of the class number, i.e., from 10 classes to 50 classes.

Table 9 Average precision and precision at top 1, 10, and 50 of all compared methods

	OMLLR Matlab	OMLLR($\gamma = 0$) Matlab	OASIS Matlab	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	Euclidean -
10 classes							
Mean avg prec.	**41** $\pm$ 1.6	34 $\pm$ 1.6	33 $\pm$ 1.6	29 $\pm$ 1.7	27 $\pm$ 0.8	24 $\pm$ 1.6	23 $\pm$ 1.9
Top 1 prec.	**51** $\pm$ 2.8	44 $\pm$ 3.2	43 $\pm$ 4.0	39 $\pm$ 5.1	39 $\pm$ 4.8	38 $\pm$ 5.4	37 $\pm$ 4.1
Top 10 prec.	**45** $\pm$ 2.2	39 $\pm$ 2.6	38 $\pm$ 1.3	33 $\pm$ 1.8	32 $\pm$ 1.2	29 $\pm$ 2.1	27 $\pm$ 1.5
Top 50 prec.	**34** $\pm$ 1.0	26 $\pm$ 1.5	23 $\pm$ 1.5	22 $\pm$ 1.3	20 $\pm$ 0.5	18 $\pm$ 1.5	18 $\pm$ 0.7
20 classes							
Mean avg prec.	**23** $\pm$ 1.3	21 $\pm$ 1.3	21 $\pm$ 1.4	17 $\pm$ 1.2	16 $\pm$ 1.2	14 $\pm$ 0.6	14 $\pm$ 0.7
Top 1 prec.	**33** $\pm$ 1.7	29 $\pm$ 1.8	29 $\pm$ 2.6	26 $\pm$ 2.3	26 $\pm$ 2.7	26 $\pm$ 3.0	25 $\pm$ 2.6
Top 10 prec.	**26** $\pm$ 1.6	23 $\pm$ 1.7	24 $\pm$ 1.9	21 $\pm$ 1.5	20 $\pm$ 1.4	19 $\pm$ 1.0	18 $\pm$ 1.0
Top 50 prec.	**20** $\pm$ 1.0	17 $\pm$ 0.6	15 $\pm$ 0.4	14 $\pm$ 0.5	13 $\pm$ 0.6	11 $\pm$ 0.2	12 $\pm$ 0.2
50 classes							
Mean avg prec.	**14** $\pm$ 0.3	13 $\pm$ 0.4	12 $\pm$ 0.4	$*$	9 $\pm$ 0.4	8 $\pm$ 0.4	9 $\pm$ 0.4
Top 1 prec.	**22** $\pm$ 1.4	18 $\pm$ 1.5	21 $\pm$ 1.6	$*$	18 $\pm$ 0.7	18 $\pm$ 1.3	17 $\pm$ 0.9
Top 10 prec.	**17** $\pm$ 0.3	15 $\pm$ 0.4	16 $\pm$ 0.4	$*$	13 $\pm$ 0.6	12 $\pm$ 0.5	13 $\pm$ 0.4
Top 50 prec.	**12** $\pm$ 0.4	11 $\pm$ 0.3	10 $\pm$ 0.3	$*$	8 $\pm$ 0.3	7 $\pm$ 0.2	8 $\pm$ 0.3

Values are averages over 5-fold cross-validations; $\pm$ values are the standard deviation across the 5 folds. A '$*$' denotes cases where a method takes more than 5 days to converge. OMLLR($\gamma = 0$) means it does not consider the low-rank constraint

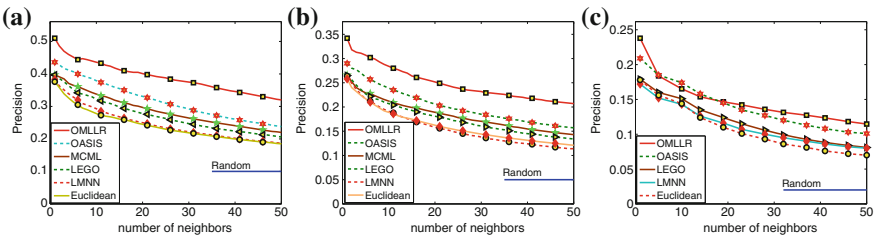

Fig. 11 Comparison of the performance of OMLLR, OASIS, LMNN, MCML, LEGO and the Euclidean metric in feature space. Each *curve* shows the precision at top k as a function of k neighbors. The results are averaged across 5 train/test partitions (40 training images, 25 test images), error bars are standard error of the means, *black dashed line* denotes chance performance. **a** 10 classes. **b** 20 classes. **c** 50 classes

This is because for a fixed training steps (35 k iterations), the more the number of classes, the lower the probability of different samples meet each other, which will destroy the performance. Figure 11 demonstrates the precision curve for retrieval, and the performance of our method is better than others for all cases.

6.6 Comparisons

i. Evaluating the effectiveness of low-rank constraint:
To justify the effectiveness of low-rank constraint, we can eliminate the impact of low-rank constraint by setting the value of γ in Eq. (6) to 0, which is similar to the model definition of Eq. (4) as OASIS. The results are shown in Table 9 using Caltech 256 dataset, the performance of our OMLLR is the best one; in comparison, ours with γ as 0 decreases accordingly and is similar to other models without low-rank constraint, such as OASIS, MCML, LEGO and LMNN. This result again justifies the effectiveness of low-rank constraint.

ii. Comparing the influence of varying the initial training data size:
We adopt the synthesized data to analyze the influence of the size of the initial training data, which varies from 100 to 1000, as shown in Table 10. In Fig. 12, we also graphic display the comparing the influence of various training data size. We can find that by increasing the size of training data from 100 to 750, the accuracy of our model OMLLR for both Ours1 and Ours2 is improved significantly; and from the case of 750–1000, as the data size is large enough, the performance of our model is not changed. For other practical applications, a larger amount of training data is helpful to improve the performance of online learning model. However, it needs more iterations and consumes more computation time. Therefore, users should balance the size of training data and computational cost.

Table 10 Comparing the influence of various training data size

	100	250	500	750	1000
Ours1	84.35	88.05	89.76	90.10	90.11
Ours2	86.35	88.44	89.76	90.11	90.12

The first row indicates the training datasize varying from 100 to 1000

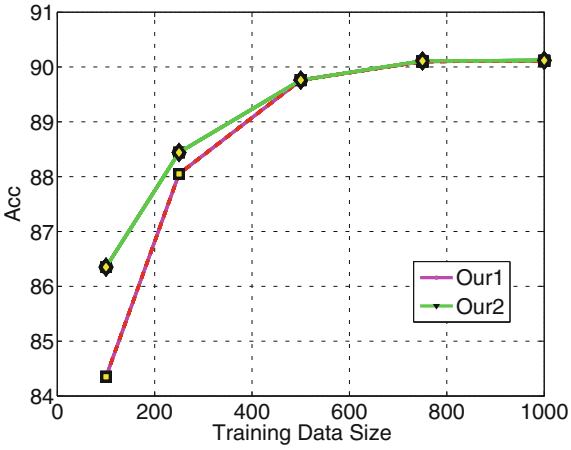

Fig. 12 Comparing the influence of various training data size. The first row indicates the training datasize varying from 100 to 1000

Table 11 Comparing the Bi-linear Graph with the other classical similarity graphs (Full Graph, K-NN Graph and $\epsilon-$NN Graph) under various parameters

σ	Full Graph	K-NN Graph				$\epsilon-$NN Graph				Bi-linear Graph
		K = 10	K = 30	K = 50	K = 100	$\epsilon = 25$	$\epsilon = 100$	$\epsilon = 900$	$\epsilon = 2500$	
$\sigma = 5$	65.1	63.7	71.7	72.8	71.5	20.7	62.1	65.1	65.1	**77.03**
$\sigma = 10$	56.5	64.0	68.6	69.1	65.8	20.7	60.4	56.5	56.5	
$\sigma = 20$	51.2	62.9	65.7	65.2	60.4	20.7	60.4	51.2	51.2	

Table 12 The comparison of time consumption, when the feature dimension increases from 40 to 200 in the top row

Method	40	80	120	160	200
Ours	0.434	1.511	3.885	7.656	12.151
OASIS [9, 47]	0.092	0.096	0.107	0.114	0.130

The time consumption for 1000 iterations is recorded accordingly

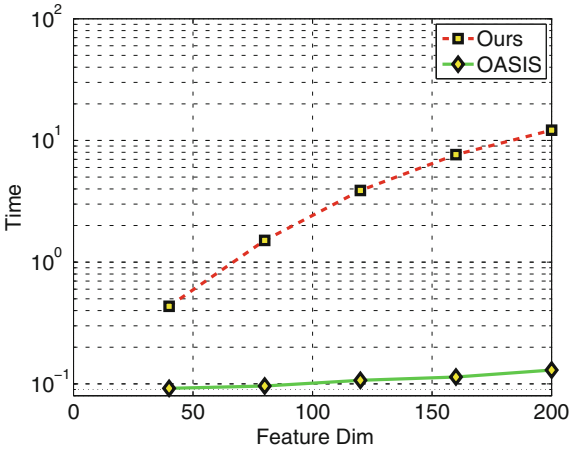

Fig. 13 The comparison of time consumption, when the feature dimension increases from 40 to 200 in the *top* row. The time consumption for 1000 iterations is recorded accordingly. The *horizontal* axis is the feature dimension and the *vertical* axis is the time consumption after log normalization

iii. Comparing Bi-linear Graph with different similarity graphs:
To validate the effectiveness of our Bi-linear graph, we compare our Bi-linear graph with the state-of-the-art graphs, such as $k-$NN, $\epsilon-$NN). The statistic results are shown in Table 11 using Sport 8 dataset. We can see that our proposed Bi-linear graph model not only outperforms other graphs, but does also not need to tune any parameters about the graph, where the traditional similarity graphs are parameter sensitive and their performances are not robust without a suitable selection of the parameters, e.g., $\sigma = 20$ or $\epsilon = 25$.

iv. Comparing the time Consumption:
In this part, we compare the time consumption of our OMLLR with the state-of-the-art methods using both the synthesized data and real data, i.e., Caltech 256. For the synthesized data, Table 12 illustrates the comparison of time consumption between our OMLLR and the classical model, OASIS [9, 47]. With the

Table 13 Runtime (minutes) of all compared methods (around 35 K training steps)

	OMLLR Matlab	OASIS Matlab	OASIS Matlab+C	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	fastLMNN Matlab+C
10 classes	342 ± 31	42 ± 15	0.12 ± 0.03	1835 ± 210	143 ± 44	337 ± 169	247 ± 209
20 classes	550 ± 43	45 ± 8	0.15 ± 0.02	7425 ± 106	533 ± 49	631 ± 40	365 ± 62
50 classes	731 ± 71	25 ± 2	1.6 ± 0.04	$*$	711 ± 28	960 ± 80	2109 ± 67

feature dimension increasing from 40 to 200, the time consumption is recorded every 1000 iterations. We also demonstrate the time consuming in Fig. 13.

The comparison of time consumption for Caltech 256 dataset is shown in Table 13, where our OMLLR is slower than OASIS, comparable with LEGO and LMNN, but much more efficiency than MCML. Even though our OMLLR is more time consuming than OASIS, the performance of our OMLLR is better than other online metric learning methods, as shown in Table 9. This is because we adopt the SVD transformation for model optimization. All the experiments

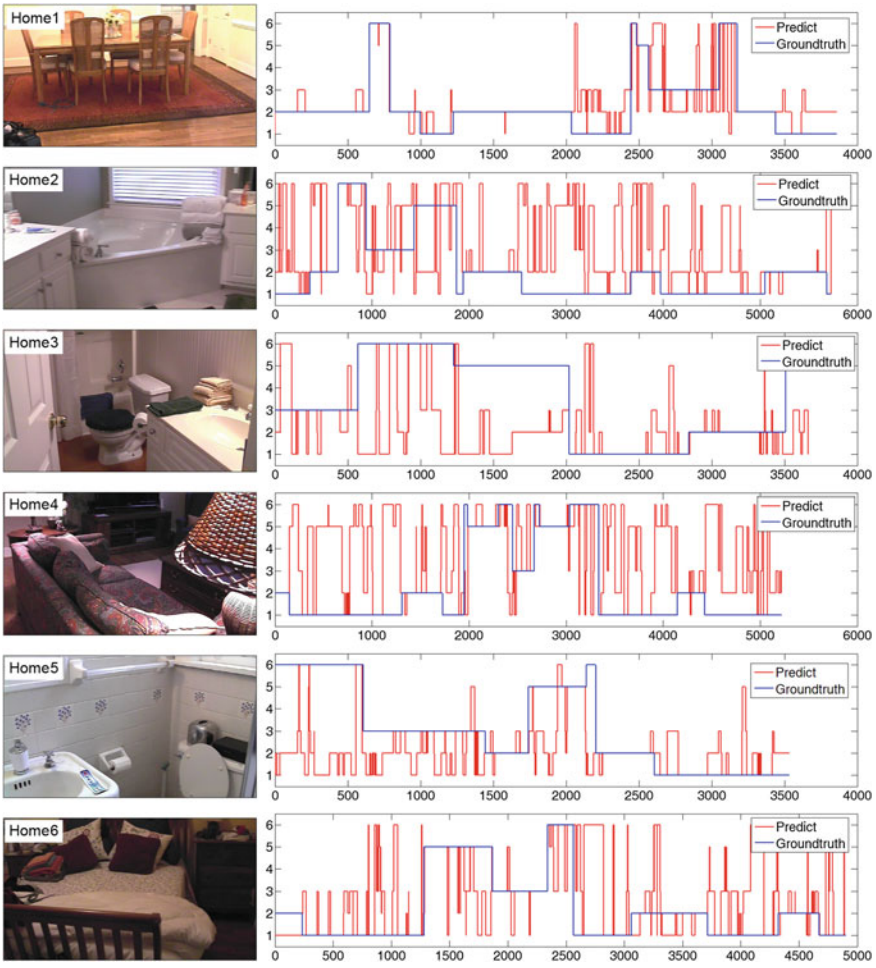

Fig. 14 The results of scene categorization for VPC 09. The images of the *left* column are examples of each home. Each figure of the *right* column is the frame-level result, where the *red* and *blue line* correspond to the predicted result of our methodology after smooth filter and the ground truth, respectively, and the x-axis is the frame index and the y-axis is the 5 class labels (bed, bath, kitchen, living and dining correspond to label 1, 2, 3, 5 and 6 respectively with label 4 absent)

are performed on the computer with 4G RAM, Pentium IV 2.6 GHz CPU. Our OMLLR is fully implemented in Matlab.

7 Summary

Most state-of-the-art scene recognition technologies rely on offline training in a batch model, thus may not be suitable for online scene recognition, which is still a challenging problem for computer vision. As the online image data characteristics may change over time, in this chapter, we present an incremental metric learning framework for self-supervised online scene classification. Given a number of labeled samples to initialize the similarity metric followed by a sequential input stream of unseen testing samples, the similarity metric is updated by maximizing the margin between different classes of samples with a low-rank constraint. The pair-wise similarity is measured by our new bi-linear graph for online label propagation to the new data. Next, by retaining the new images that are confidently labeled, the scene recognition model is further updated. Experiments on various benchmark datasets and comparisons with other state-of-the-art methods demonstrate the effectiveness and efficiency of our algorithm. Besides online scene recognition, our proposed online learning framework can also be applied to other applications, such as object detection [53], object tracking [54], and image retrieval [9].

Acknowledgments This work was supported in part by Natural Science Foundation of China (61105013, 61375014).

Appendix

Proof of Theorem 1

Proof Since W is a PSD matrix, it can be decomposed as $W = UU^T$ where $U \in \mathbb{R}^{d\times d}$. Consider the following equation $X^TV = X^TU$ with respect to V. Define $B \in \mathbb{R}^{d\times(d-r)}$ with linear dependent columns $B_{.i}$'s in the null space of X^T. One can obtain the solution as $V = U + BZ$ where $Z \in \mathbb{R}^{(d-r)\times d}$. Split U and B into two parts $U = \binom{U1}{U2}$ and $B = \binom{B1}{B2}$ where $U1 \in \mathbb{R}^{(d-r)\times d}$, $U2 \in \mathbb{R}^{r\times d}$, $B1 \in \mathbb{R}^{(d-r)\times(d-r)}$, and $B2 \in \mathbb{R}^{r\times r}$. Define $Z = -B1^{-1}U1$. One verifies that $V = \binom{0}{U2-B2B1^{-1}U1}$ and its rank is at most r. Since $X^TU = X^TV$, we obtain $X^TWX = X^TQX$ and the rank of Q is r by letting $Q = VV^T$. ■

Proof of Theorem 2

Proof Decompose C into the symmetric space and the skew symmetric space, i.e., $C = C_y + C_k$ where $C_y = \frac{1}{2}(C + C^T)$ and $C_k = \frac{1}{2}(C - C^T)$. Note that $\langle C_y, C_k \rangle = 0$. Consider $W \succeq 0$ (W must be symmetric) in the following

$$\begin{aligned} \|W - C\|_F^2 &= \|W - C_y - C_k\|_F^2 \\ &= \|W - C_y\|_F^2 + \|C_k\|_F^2 + 2\langle W - C_y, C_k \rangle \\ &= \|W - C_y\|_F^2 + \|C_k\|_F^2. \end{aligned} \tag{26}$$

Thus, we obtain $prox_{\gamma P,\Omega}(C) = prox_{\gamma P,\Omega}(C_y)$.

$$\begin{aligned} &\min_{W \succeq 0} \frac{1}{2}\|W - C_y\|_F^2 + \gamma \|W\|_* \\ &= \min_{W \succeq 0} \frac{1}{2}\|W - C_y\|_F^2 + \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \langle W, Z \rangle \\ &= \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \min_{W \succeq 0} \frac{1}{2}\|W - C_y\|_F^2 + \langle W, Z \rangle \\ &= \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \min_{W \succeq 0} \frac{1}{2}\|W - C_y + Z\|_F^2 + \langle C_y, Z \rangle - \frac{1}{2}\|Z\|_F^2 \\ &= \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \frac{1}{2}\|(C_y - Z)^-\|_F^2 + \langle C_y, Z \rangle - \frac{1}{2}\|Z\|_F^2 \end{aligned} \tag{27}$$

The first equality uses the dual form of the trace norm of a PSD matrix, where $S\mathbb{R}$ denotes the symmetric space. The second equality is due to Von Neumann theorem. The last equality uses the result that the projection from a symmetric matrix X onto the SDP cone is X^+, which also implies that $W = (C_y - Z)^+$.

It follows that

$$\begin{aligned} &\max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \frac{1}{2}\|(C_y + Z)^-\|_F^2 + \langle C_y, Z \rangle - \frac{1}{2}\|Z\|_F^2 \\ &= \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} \frac{1}{2}\|(C_y - Z)^-\|_F^2 - \frac{1}{2}\|C_y - Z\|_F^2 + \frac{1}{2}\|C_y\|_F^2 \\ &= \max_{\|Z\| \leq \gamma, Z \in S\mathbb{R}^{d \times d}} -\frac{1}{2}\|(C_y - Z)^+\|_F^2 + \frac{1}{2}\|C_y\|_F^2 \end{aligned} \tag{28}$$

From the last formulation, we obtain the optimal $Z^* = \mathcal{T}_\gamma(C_y)$ and the optimal $W^* = (C_y - Z^*)^+ = (C_y - \mathcal{T}_\gamma(C_y))^+ = \mathcal{D}_\gamma(C_y)^+ = D_\gamma(C_y)$. It completes our proof. ■

References

1. F. Perronnin, Z. Akata, Z. Harchaoui, C. Schmid, Towards good practice in large-scale learning for image classification, in *CVPR IEEE*, pp. 3482–3489 (2012)
2. S. McCann, D.G. Lowe, Local naive bayes nearest neighbor for image classification, in *CVPR IEEE*, pp. 3650–3656 (2012)
3. B. Fernando, E. Fromont, T. Tuytelaars, Mining mid-level features for image classification. Int. J. Comput. Vision, pp. 1–18 (2014)
4. J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice. Int. J. Comput. Vision **105**(3), 222–245 (2013)
5. O. Russakovsky, Y. Lin, K. Yu, L. Fei-Fei, Object-centric spatial pooling for image classification, in *Computer Vision-ECCV*. (Springer), 1–15 (2012)
6. K. Simonyan, A. Vedaldi, A. Zisserman, Deep fisher networks for large-scale image classification, in *Advances in Neural Information Processing Systems*, pp. 163–171 (2013)
7. H. Kekre, S. Thepade, R. K. K. Das, S. Ghosh, Image classification using block truncation coding with assorted color spaces. Int. J. Comput. Appl., vol. 44 (2012)
8. H. Grabner, H. Bischof, On-line boosting and vision, in *CVPR*, vol. 1, pp. 260–267 (2006)
9. G. Chechik, V. Sharma, U. Shalit, S. Bengio, An online algorithm for large scale image similarity learning. NIPS **21**, 306–314 (2009)
10. F. Wang, C. Yuan, X. Xu, P. van Beek, Supervised and semi-supervised online boosting tree for industrial machine vision application, in *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data. ACM*, pp. 43–51 (2011)
11. G. Cauwenberghs, T. Poggio, Incremental and decremental support vector machine learning, in *NIPS*, pp. 409–415 (2001)
12. B. Liu, S. Mahadevan, J. Liu, Regularized off-policy td-learning, in *NIPS* (2012)
13. Y. Cong, J. Yuan, Y. Tang, Object tracking via online metric learning, in *ICIP*, pp. 417–420 (2012)
14. Y. Cong, J. Liu, J. Yuan, J. Luo, Self-supervised Online Metric Learning with Low Rank Constraint for Scene Categorization. IEEE Trans. Image Process. **22**(8), 3179–3191 (2013)
15. K. van de Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1582–1596 (2010)
16. M. Szummer, R. Picard, Indoor-outdoor image classification, in *IEEE International Workshop on Content-Based Access of Image and Video Database* (1998), pp. 42–51
17. D. Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
18. A.E. Abdel-Hakim, A.A. Farag, Csift: a sift descriptor with color invariant characteristics, *CVPR*, vol. 2006 (IEEE, 2006), pp. 1978–1983
19. A. Bosch, A. Zisserman, X. Muoz, Scene classification using a hybrid generative/discriminative approach. Pattern Anal. Mach. Intell, IEEE Trans. **30**(4), 712–727 (2008)
20. M. Brown, S. Susstrunk, Multi-spectral sift for scene category recognition, in *CVPR* (IEEE, 2011), pp. 177–184 (2011)
21. J. Van De Weijer, T. Gevers, A.D. Bagdanov, Boosting color saliency in image feature detection. Pattern Anal. Mach. Intell, IEEE Trans. **28**(1), 150–156 (2006)
22. S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in *CVPR*, vol. 2 (2006)
23. J.V. Gemert, J. Geusebroek, C. Veenman, A. Smeulders, Kernel codebooks for scene categorization, in *ECCV*, pp. 696–709 (2008)
24. A. Quattoni, A. Torralba, Recognizing indoor scenes, in *CVPR* (2009)
25. J. Wu, H. Christensen, J. Rehg, Visual place categorization: Problem, dataset, and algorithm, in *IROS* (2009)
26. J. Wu, J. Rehg, CENTRIST: A visual descriptor for scene categorization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1489–1501 (2011)
27. Y. Xiao, J. Wu, J. Yuan, mcentrist: A multi-channel feature generation mechanism for scene categorization (2014)

28. Y. Cong, J. Yuan, J. Luo, Towards scalable summarization of consumer videos via sparse dictionary selection. Multimedia, IEEE Trans. **14**(1), 66–75 (2012)
29. L. Li, L. Fei-Fei, What, where and who? Classifying events by scene and object recognition, in *ICCV*, vol. **2**(4), 8 (2007)
30. D. Walther, E. Caddigan, L. Fei-Fei, D. Beck, Natural scene categories revealed in distributed patterns of activity in the human brain. J. Neurosci. **29**(34), 10573 (2009)
31. L. Li, R. Socher, L. Fei-Fei, Towards total scene understanding: classification, annotation and segmentation in an automatic framework, in *CVPR*, pp. 2036–2043 (2009)
32. J. Liu, M. Shah, Scene modeling using co-clustering, in *ICCV* (2007)
33. P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, A thousand words in a scene. IEEE Trans. Pattern Anal. Mach. Intell. **29**(9), 1575–1589 (2007)
34. A. Bosch, A. Zisserman, M. Pujol, Scene classification using a hybrid generative/discriminative approach, IEEE transactions on pattern analysis and machine intelligence, vol. 30 (2008)
35. J. Kivinen, E. Sudderth, M. Jordan, Learning multiscale representations of natural scenes using Dirichlet processes, in *ICCV* (2007)
36. J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval. Int. J. Comput. Vision **72**(2), 133–157 (2007)
37. P. Utgoff, N. Berkman, J. Clouse, Decision tree induction based on efficient tree restructuring. Mach. Learn **29**(1), 5–44 (1997)
38. S. Avidan, Ensemble tracking, in *CVPR*, vol. 2, pp. 494–501 (2005)
39. X. Liu, T. Yu, Gradient feature selection for online boosting, in *ICCV*, pp. 1–8 (2007)
40. N. Oza, S. Russell, Online bagging and boosting, in *Artif. Intell. Stat* (2001)
41. E. Lughofer, On-line evolving image classifiers and their application to surface inspection. Image Vis. Comput. **28**(7), 1065–1079 (2010)
42. F. Gayubo, J. Gonzalez, E.D.L. Fuente, F. Miguel, J. Peran, On-line machine vision system for detect split defects in sheet-metal forming processes, in *ICPR*, vol. 1, pp. 723–726 (2006)
43. O. Camoglu, T. Yu, L. Bertelli, D. Vu, V. Muralidharan, S. Gokturk, An efficient fashion-driven learning approach to model user preferences in on-line shopping scenarios, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), pp. 28–34 (2010)
44. K. Weinberger, L. Saul, Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
45. P. Jain, B. Kulis, I. Dhillon, K. Grauman, Online metric learning and fast similarity search, in *NIPS*, pp. 761–768 (2008)
46. A. Globerson, S. Roweis., Metric learning by collapsing classes, in *NIPS* (2006)
47. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive-aggressive algorithms. J. Mach. Learn. Res. **7**, 585 (2006)
48. G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking. J. Mach. Learn. Res. **11**, 1109–1135 (2010)
49. H. Cheng, Z. Liu, J. Yang, Sparsity Induced Similarity Measure for Label Propagation, in *ICCV* (2009)
50. K.Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in *NIPS* (2006)
51. "http://categorizingplaces.com/dataset.html"
52. G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset(2007)
53. N. Jacobson, Y. Freund, T. Nguyen, An online learning approach to occlusion boundary detection. Image Proc. IEEE Trans. **99**, 1–1 (2010)
54. Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, L. Bai, Real-time probabilistic covariance tracking with efficient model update. Image Proc. IEEE Trans. **21**(5), 2824–2837 (2012)