Online Similarity Learning for Big Data with Overfitting

Yang Cong, *Senior Member, IEEE*, Ji Liu, Baojie Fan, Peng Zeng, Haibin Yu, and Jiebo Luo, *Fellow, IEEE*

Abstract—In this paper, we propose a general model to address the overfitting problem in online similarity learning for big data, which is generally generated by two kinds of redundancies: 1) feature redundancy, that is there exists redundant (irrelevant) features in the training data; 2) rank redundancy, that is non-redundant (or relevant) features lie in a low rank space. To overcome these, our model is designed to obtain a simple and robust metric matrix through detecting the redundant rows and columns in the metric matrix and constraining the remaining matrix to a low rank space. To reduce feature redundancy, we employ the group sparsity regularization, i.e., the $\ell_{2,1}$ norm, to encourage a sparse feature set. To address rank redundancy, we adopt the low rank regularization, the *m*ax norm, instead of calculating the SVD as in traditional models using the nuclear norm. Therefore, our model can not only generate a low rank metric matrix to avoid overfitting, but also achieves feature selection simultaneously. For model optimization, an online algorithm based on the stochastic proximal method is derived to solve this problem efficiently with the complexity of $O(d^2)$. To validate the effectiveness and efficiency of our algorithms, we apply our model to online scene categorization and synthesized data and conduct experiments on various benchmark datasets with comparisons to several state-of-the-art methods. Our model is as efficient as the fastest online similarity learning model OASIS, while performing generally as well as the accurate model OMLLR. Moreover, our model can exclude irrelevant / redundant feature dimension simultaneously.

Index Terms—Online learning, similarity learning, low rank, sparse representation, feature selection, overfitting, redundancy, big data

1 Introduction

An appropriate similarity measure [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] is one of the key issues in many computer vision problems. Compared with using traditional fixed metrics, e.g., euclidean and Mahalanobis distance. Similarity learning models including both online and offline learning [11], [12], [13], [14], [15] can lead to more meaningful distance metrics automatically, and usually learn a function $S_W(p_1, p_2) = p_1^T W p_2$ [16], [17] with a bilinear form parameterized by a metric matrix $W \in \mathbb{R}^{d\times d}$ to measure the similarity between any two features $p_1, p_2 \in \mathbb{R}^d$. Intuitively, $S_W(p_1, p_2)$ assigns high scores if p_1 and p_2 are similar or from the same class, and vice versa. Specifically, similarity learning with the matrix W as a positive semi-definite matrix is also named as metric learning.

In this paper, we focus on online similarity learning [16], [18], [19], [20], [21], which learns from one or a small number of instances per iteration and provides an efficient way to incorporate new incoming data. Because the testing data always comes sequentially in practice, the performance of offline similarity learning models with batch training may deteriorate over time as the new incoming data may deviate from the initial training data. In contrast, online similarity learning has more advantages. However, the "Overfitting" phenomenon always exists: the training data that can be fitted by a simple model but instead is fitted by an unnecessarily complicate model. Although the complicate model can explain the training data as well as the simple model, its performance on the testing data is usually much worse due to poor generalizability. Therefore, the simplest model is always preferred in practice. In this work, we wish to learn a more dense submatrix W' by removing the redundant features. There are two main types of overfitting issues in online similarity learning:

a. Feature redundancy: Because we do not have enough prior knowledge to design the most effective features, there often exist redundant or even irrelevant features, which not only increases the computation cost, but also degrades the classification performance especially when the noisy level is high.
b. Rank redundancy: If the relevant features with dimension d reside in a low dimensional subspace r ($r < d$), a metric matrix with the rank lower than r can distinguish any two samples [21, Theorem 1]. Therefore, a simple low rank model can be less sensitive to noise data and more robust against overfitting.

To address these two issues, we formulate online similarity learning as an optimization problem by considering two regularization terms: a) the $\ell_{2,1}$ norm to restrict the feature

- *Y. Cong is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and the Department of Computer Science, University of Rochester, Rochester, NY 14611. E-mail: congyang81@gmail.com.*
- *J. Liu and J. Luo are with the Department of Computer Science, University of Rochester, Rochester, NY 14611. E-mail: {jliu, jiebo.luo}@cs.rochester.edu.*
- *B. Fan is with the College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210042, China. E-mail: jobfbj@gmail.com.*
- *P. Zeng is with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. E-mail: zp@sia.cn.*
- *H. Yu is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. E-mail: yhb@sia.cn.*

Manuscript received 12 June 2016; revised 1 Mar. 2017; accepted 24 Mar. 2017. Date of publication 27 Mar. 2017; date of current version 7 Mar. 2018. (Corresponding author: Yang Cong.)
Recommended for acceptance by H. Xiong.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TBDATA.2017.2688360

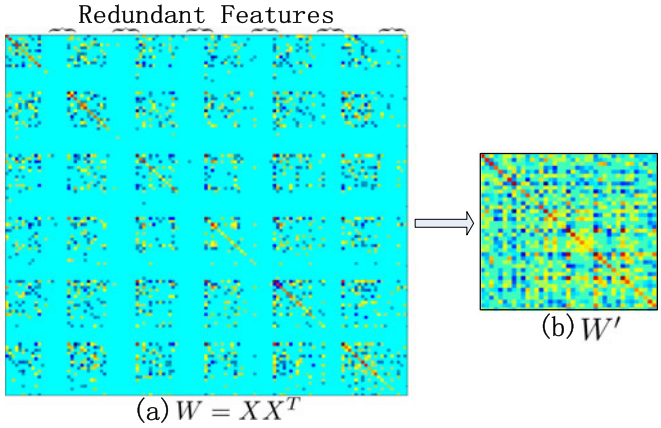


Fig. 1. An illustrative example: (a) The symmetric matrix W with light blue entries corresponding to small but nonzero values, has various redundancy. This would cause overfitting issue because these irrelevant features are not removed. (b) The proposed method allows to remove irrelevant features and compress redundant features into a smaller, simpler matrix W', which reduces overfitting and improves computational efficiency.

sparsity and address the feature redundancy, b) the regularization "max norm" [22], rather than the well known nuclear norm, as the low rank penalty to address the rank redundancy (the nuclear norm requires the SVD calculation for optimization and thus is not suitable for large scale data). In contrast, the max norm behaves the same as the nuclear norm to emphasize the low rank property on matrices, yet has a lower computational complexity ($O(d^2)$ versus $O(d^3)$). Fig. 1 is the demonstration of our method, where the metric matrix W learned by our model contains zero columns and rows. We can remove the redundant features (i.e., zero columns and rows from W) and reserve the condense ones. Generally, the main contributions of this paper are as follows:

i. With the help of both the $\ell_{2,1}$ norm and the max norm, we design a general framework, referred to as Online Similarity Learning with Low Rank and Group Sparsity (OSLLR-GS), to address the overfitting problem caused by various redundancies.
ii. For model optimization, we derive an online algorithm based on a stochastic optimization technique, which not only leads to a closed form solution for each online iteration, but also reduces the computation burden from $O(d^3)$ to $O(d^2)$ by avoiding the SVD calculation used by the traditional methods.
iii. We design extensive simulated and real experiments to validate the effectiveness of our model. Its performance is comparable to the accurate OMLLR [21] model and is as fast as the OASIS model [16], [23].

The rest of the paper is organized as follows. In Section 2, we review the related works. Sections 3 and 4 introduce our model and the online model optimization, respectively. In Section 5, we compare our model with the state-of-the-arts. We conclude the paper in Section 6.

2 Related Works

Similarity measurement is a fundamental problem in machine learning and computer vision domain [6], [24], [25], [26], [27], [28], [29], [30]. In comparison with traditional fixed metrics, e.g., euclidean and Mahalanobis distance, metric learning [1], [2], [3], [4] is the task of learning a distance function over objects. The metric or distance function has to obey four axioms: non-negativity, identity of indiscernible, symmetry and triangle inequality constraints. There are also some Matlab toolboxes for distance metric learning[1] [1], [2]. Generally, depending on the learning paradigm, the metric learning models can be classified as fully *supervised metric learning*, *weakly supervised metric learning*, *semi-supervised metric learning* and *unsupervised distance metric learning* (e.g., Principal Component Analysis (PCA) and Locally Linear Embedding (LLE) [31]); based on the form of metric, the metric learning models have *linear*, *nonlinear* and *local variations*; depending on the optimality of the solution, they have *local* and *global* optimal solutions; and there are also *online* and *batch* metric learning models. In this paper, we mainly focus on *supervised online metric learning models*. For example, the large margin nearest neighbor method (LMNN) [18] is proposed to learn a Mahalanobis distance such that the k-nearest neighbors of a given sample belong to the same class and different-class samples are separated by a large margin. LEGO [19], online learning of a Mahalanobis distance using a Log-Det regularization per instance loss, is guaranteed to yield a positive semidefinite matrix. In [20], a metric learning algorithm by collapsing classes (MCML) is designed to learn a Mahalanobis distance such that same-class samples are mapped to the same point; however, the MCML model is too time consuming. Chechik et al. [16], [17], [23] design an Online Algorithm for Scalable Image Similarity learning (OASIS) for learning pairwise similarity. OASIS is fast and scales linearly with the number of objects and the number of non-zero features, but it may suffer from overfitting. Ying et al. [32] intend to pursue the metric learning model based on sparse representation. Jin et al. [33] present an efficient online learning algorithm for regularized distance metric learning (RDML). Bian et al. [34] propose a constrained empirical risk minimization framework for distance metric learning (CDML). Kunapuli et al. [35] propose an online regularized metric learning algorithm based on composite objective mirror descent (COMID). Huang et al. [36] intend to propose a unified framework for sparse metric learning. Most of the above metric learning models have not considered the low rank property of the real data.

To avoid overfitting problem, the low rank constraint has been considered into online metric learning [21], [37], [38], [39], [40]. Liu et al. [40] propose an interesting work to handle the similarity metric learning with low-rank constraint for high dimensional data, where the number of training samples n is much smaller than the feature dimension d, i.e., $n << d$. Mei et al. [27] propose a logdet divergence-based metric learning model. In [38], [39], Shalit et al. design an embedded manifold of low rank matrix for online metric learning model, and in [37], a Riemannian method is proposed to pursue a low rank positive semidefinite matrix. [5] theoretically analyses the Learning Vector Quantization (LVQ) using a quadratic matrix of adaptive relevance parameters to pursue the low rank property. In our previous work [21], we propose an online similarity learning model via low rank constraint, which adopts the nuclear

1. http://www.cs.cmu.edu/liuy/distlearn.htm

norm regularization to pursue the low rank property of the model matrix and can partially overcome overfitting caused by rank redundancy. In comparison with [37], [38], [39], we pursue the minimum expectation as the loss function, which is more robust. However, our previous OMLLR needs to calculate the time consuming SVD in each iteration. Therefore, our new model avoids the SVD calculation, which is more efficient than OMLLR (i.e., the computational complexity decreasing from $O(d^3)$ to $O(d^2)$; moreover, ours can pursue both adaptive distance metric learning and feature selection concurrently. The most similar work is [41], which proposes a novel algorithm to pursue the low rank and group sparsity structures by adopting the nuclear norm and $\ell_{2,1}$ norm on the metric matrix W. However, [41] requires extensive computation burden (eigenvalue decomposition for PSD constraint and ADMM framework to separate $\ell_{2,1}$ norm and nuclear norm) per iteration to solve the formulation; in comparison, we enforce low rank structure using max norm and group sparsity structure using $\ell_{2,1}$ norm on X ($W = XX^T$) instead of W, which leads to a much simpler and more efficient optimization algorithm. Therefore, our proposed algorithm is more suitable for the online and large case learning scenarios.

3 The Proposed Online Similarity Learning Model

This section introduces our similarity learning model to handle overfitting mainly caused by two types of redundancy, i.e., feature redundancy and rank redundancy. Our online similarity learning model contains two parts, the loss function and regularization terms. For the loss function, we calculate the expectation of the model for all the data, which can generate a robust result. For the regularization terms, in comparison with most previous works pursuing the low rank constraint [21], [37], [38], [39], we consider an additional group sparse constraint to achieve feature selection. Therefore, our model can not only calculate the metric matrix in a low rank space, but also detects the redundant rows and columns simultaneously, i.e., our model can pursue a simple and robust metric matrix with low rank and feature sparsity.

3.1 Loss Function

We define the similarity metric function as

$$S_W(p_1, p_2) = p_1^T W p_2. \tag{1}$$

Intuitively, $S_W(p_1, p_2)$ assigns high scores if p_1 and p_2 are similar or from the same class, and vice versa. For robustness, a soft margin is often used

$$S_W(p_1, p_2) \geq S_W(p_2, p_3) + 1, \tag{2}$$

where p_1 is from the same class as p_2 and p_3 is from a different one. One can use various loss functions to penalize a violation, such as hinge loss and logistic loss. Throughout this paper, we use the hinge loss function

$$l(W; p1, p2, p3) = \max\{0, 1 - p_1^T W p_2 + p_2^T W p_3\}. \tag{3}$$

Now we are ready to introduce the loss function. T is a set of tuples and each tuple contains three indexes $\{t_1, t_2, t_3\}$ (similar technology was also used in WSABIE [42]) where p_{t_1} and p_{t_2} are from the same class, which is different from the class of p_{t_3}. We define the average loss as the loss function

$$\frac{1}{|T|} \sum_{\{t_1,t_2,t_3\} \in T} l(W; p_{t_1}, p_{t_2}, p_{t_3}), \tag{4}$$

where $|T|$ denotes the size of T. If the training data contains multiple classes, T would be all possible tuples satisfying the condition above. Note that the size of T is on the order of $O(n^3)$ (n is the size of training data). Let $t = \{t_1, t_2, t_3\}$ and $l_t(W) := l(W; p_{t_1}, p_{t_2}, p_{t_3})$. Actually, the loss function Eq. (4) can be considered as the expectation of the loss, which is more robust than the loss function in [16], [23]; and we can simply rewrite it as

$$\mathbb{E}[l_t(W)] = \frac{1}{|T|} \sum_{\{t_1,t_2,t_3\} \in T} l(W; p_{t_1}, p_{t_2}, p_{t_3}). \tag{5}$$

3.2 Low Rank Regularization and Feature Sparsity Regularization

The nuclear norm of matrices has been widely used to enforce the rank sparsity on matrices. Its main disadvantage is that it requires the SVD calculation, which is quite expensive in computation and cannot handle large scale matrices. To avoid computing SVD, we apply the low rank regularization, *max* norm. The *max* norm has been introduced in machine learning community for matrix reconstruction [43] and clustering [44]. The max norm of matrices is defined as

$$\begin{aligned} \|W\|_{\max} &:= \inf_{W=LR^T} \{\|L\|_{2,\infty} \|R\|_{2,\infty}\} \\ &= \inf_{W=LR^T, \|L\|_{2,\infty} = \|R\|_{2,\infty}} \{\|L\|_{2,\infty} \|R\|_{2,\infty}\} \\ &= \inf_{W=LR^T, \|L\|_{2,\infty} = \|R\|_{2,\infty}} \max\{\|L\|_{2,\infty}^2, \|R\|_{2,\infty}^2\} \\ &= \inf_{W=LR^T} \max\{\|L\|_{2,\infty}^2, \|R\|_{2,\infty}^2\}, \end{aligned} \tag{6}$$

where the $\ell_{2,\infty}$ norm is defined by $\|X\|_{2,\infty} := \max_{k=1}^{d} \|X_{k.}\|$ ($X_{k.}$ is the kth row of X), $L \in \mathbb{R}^{d \times m}$ and $R \in \mathbb{R}^{d \times m}$. To show why the max norm enforces the low rank property like the nuclear norm, we have the following equations:

$$\|W\|_* = \inf_{\|u\|=1, \|v\|=1} \left\{ \sum_i |\sigma_i| \; | W = \sum_{i=1} \sigma_i u_i v_i^T \right\}, \tag{7}$$

$$\|W\|_{\max} = K \inf_{\|u\|_\infty=1, \|v\|_\infty=1} \left\{ \sum_i |\sigma_i| \; | W = \sum_{i=1} \sigma_i u_i v_i^T \right\}, \tag{8}$$

where $\|.\|_*$ denotes the nuclear norm. The range of the equivalence factor K is $[1.676, 1.738]$ [22], [45]. The two equivalent forms above imply that the max norm enforces the rank sparsity with factors in the ℓ_∞ space while the nuclear norm does so in the ℓ_2 space. The main advantage of *max norm* over nuclear norm is lower computation cost, which will become clear in Section 4.

In our problem, to enforce the rank sparsity property, it is natural to model our problem as

$$\begin{aligned} \min_{X} &: \mathbb{E}[l_t(W)], \\ s.t. &: \|W\|_{\max} \leq \lambda^2, \\ & W \succeq 0. \end{aligned} \tag{9}$$

Since W is constrained in the positive semi-definite cone, its max norm can also be expressed as $\|W\|_{\max} = \{\|X\|_{2,\infty}^2 | W = XX^T\}$. λ is the tuning parameter ($\lambda \geq 0$). Using the decomposition of W, we can reformulate Eq. (9) as

$$\begin{aligned} \min_{X \in \mathbb{R}^{d \times m}} &: \mathbb{E}[l_t(W)], \\ s.t. &: \|X\|_{2,\infty}^2 \leq \lambda^2, \\ & W = XX^T. \end{aligned} \tag{10}$$

Note that the positive semi-definite constraint is automatically satisfied. This is a noncovnex problem in general because of the decomposition constraint $W = XX^T$, which can easily lead to a local optimum. Fortunately, Burer and Monteiro [46] proved that when X contains sufficient columns, this decomposition can still lead to a global solution for the symmetric, positive semidefinite variable $W = XX^T$. In our experiment, we always choose a large number of columns for X. Note that choosing a small number of columns for X can enforce a rigid low rank constraint as well, which has been reported extensively in early literature but tends to fall into a local optimum. In comparison, we choose a large value for m, which can lead to a global optimum, and use the max norm to enforce the low rank property. In other words, we can simplify the problem of Eq. (10) by removing W

$$\begin{aligned} \min_{X} &: \mathbb{E}[l_t(XX^T)], \\ s.t. &: \|X\|_{2,\infty} \leq \lambda. \end{aligned} \tag{11}$$

In addition to the low rank property for the metric matrix, we also like to select relevant features or detect irrelevant features. Each row of X corresponds to a feature. To pursue sparse features, a natural idea is to use the $\ell_{2,1}$ norm on X to enforce group sparsity, that is, we expect many rows of X to be zeros. If a whole row of X is zero, then the corresponding features is detected as irrelevant. The final objective of our online similarity learning model can be formulated as

$$\begin{aligned} \min_{X} &: F(X) : = \mathbb{E}[l_t(XX^T)] + \mu\|X\|_{2,1}, \\ & s.t. : \|X\|_{2,\infty} \leq \lambda. \end{aligned} \tag{12}$$

where the $\ell_{2,1}$ norm is defined by $\|X\|_{2,1} := \sum_{k=1}^{d} \|X_{k.}\|_2$, and μ is the tuning parameter ($\mu \geq 0$). We will pursue a fast model optimization of Eq. (12) in next section.

4 Model Optimization

This section describes how to solve the optimization problem in Eq. (12). The traditional offline (or batch) methods require the evaluation of the full gradient of the loss function $\mathbb{E}[l_t(XX^T)]$ iteratively, thus leading to a heavy computation load when the size of T is too large. In order to solve the optimization problem in Eq. (12) efficiently, we use the stochastic proximal gradient descent method [47], which only uses a very small number of training samples for model updating in each iteration. The stochastic proximal gradient descent method updates the $(i+1)$th iteration by

$$\begin{aligned} X^{i+1} = arg \min_{X} &: \frac{1}{2}\|X - \alpha^i G_t(X^i)\|^2 + \alpha^i \mu \|X\|_{2,1}, \\ & s.t. : \|X\|_{2,\infty} \leq \lambda, \end{aligned} \tag{13}$$

where $G_t(X^i)$ is a randomly generated subgradient satisfying $\mathbb{E}(G_t(X^i)) \in \partial \mathbb{E}[l_t(X^i(X^i)^T)]$ and $\partial f(.)$ denotes the subdifferential of $f(.)$. There are many choices of $G_t(X^i)$ to satisfy this condition. As we have

$$\partial \mathbb{E}[l_t(X^i(X^i)^T)] = \mathbb{E}[\partial l_t(X^i(X^i)^T)], \tag{14}$$

in our case, we simply choose $G_t(X^i) \in \partial l_t(X^i(X^i)^T)$ by uniformly sampling t from T, which is a typical way to sample the stochastic subgradient. If $l_t(.)$ is the hinge loss, $G_t(.)$ can be

$$G_t(X^i) = \begin{cases} (p_{t_3} - p_{t_1}) p_{t_2}^T X^i, & l_t(X^i X^{iT}) \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The subproblem in Eq. (13) is convex but includes a nonsmooth term $\|X\|_{2,1}$ and a constraint $\|X\|_{2,\infty}$. Typically, one can only obtain an approximate solution of Eq. (13), which can be extremely close to the true optimal solution (depending on how many iterations) for some state-of-the-art methods. Fortunately, since the $\|.\|_{2,\infty}$ and $\|.\|_{2,1}$ are conjugate to each other, the problem of Eq. (13) has a closed form solution shown in Theorem 1.

Algorithm 1. Online Similarity Learning via Low Rank and Group Sparsity (OSLLR-GS)

Input: $\lambda(\geq 0)$, $\mu(\geq 0)$, $c(>0)$, $m(>0)$, and T.
Output: X

1: Initialize $i = 0$ and $X_0 = \mathbf{0} \in \mathbb{R}^{d \times m}$
2: **while** True **do**
3: Let $\alpha^i = \frac{c}{\sqrt{i}}$
4: Uniformly sample t from T
5: $X_{k.}^{i+\frac{1}{2}} = X_{k.}^i - \alpha^i \partial l_t(X^i X^{iT})$
6: $L = \min\left(\frac{\lambda}{\|X_{k.}^{i+\frac{1}{2}}\|}, \max\left(1 - \frac{\alpha^i \mu}{\|X_{k.}^{i+\frac{1}{2}}\|}, 0\right)\right)$
7: $X_{k.}^{i+1} = L X_{k.}^{i+\frac{1}{2}}$
8: Update $\bar{X}^i$ by Eq. (18)
9: $i = i + 1$
10: **end while**

Theorem 1. *The optimal solution to (16)*

$$\begin{aligned} \min_{X} &: \frac{1}{2}\|X - C\|^2 + \mu\|X\|_{2,1}, \\ & s.t. : \|X\|_{2,\infty} \leq \lambda, \end{aligned} \tag{16}$$

is for all $k = 1, 2, \ldots, d$,

$$X_{k.}^* = \min\left(\frac{\lambda}{\|C_{k.}\|}, \max\left(1 - \frac{\mu\alpha^i}{\|C_{k.}\|}, 0\right)\right) C_k, \tag{17}$$

where we set $\alpha^i = \frac{c}{\sqrt{i}}$ ($c = 1$ *in our case). The proof to Theorem 1 is provided in Appendix A. Finally, we summarize the stochastic proximal algorithm for Eq. (12) in Algorithm 1.*

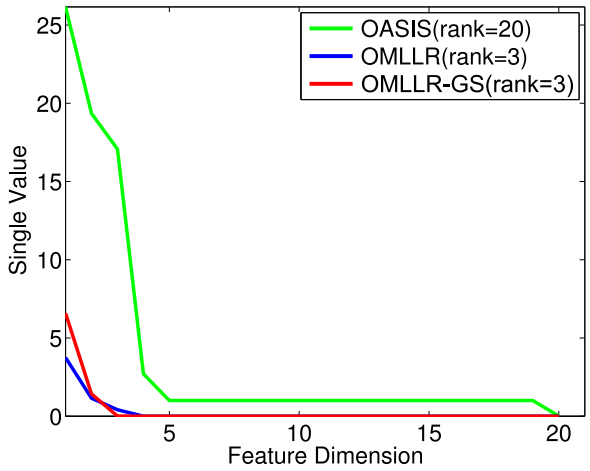


Fig. 2. Comparison of the singular values for different models with or without the low rank constraint. The horizontal axis and the vertical axis are the feature dimension and singular value, respectively.

For the general stochastic proximal gradient descent method, there is no guarantee that the sequence $\{\mathbb{E}(F(X^i))\}$ *converges. The convergent sequence is* $\{\mathbb{E}(F(\bar{X}^i))\}$ *with a convergence rate* $O(i^{-1/2})$ *where* $\bar{X}_i$ *is the weighted average of all historical iterations defined as*

$$\bar{X}^i := \left(\sum_{j=1}^{i} \alpha^j\right)^{-1} \sum_{j=1}^{i} \alpha^j X^j, \tag{18}$$

where we consider $\bar{X}$ *as our final solution due to it is more stable in practice. Actually, for space limitation we omit some additional conditions which are required to guarantee this convergence and convergence rate. Readers who are interested in it may refer to some recent literatures about stochastic optimization, e.g., [47], [48] for convex optimization, and [49], [50] for nonconvex cases.*

5 Experiments

In this section, we present several experiments and comparisons to validate the proposed model. Experiments are conducted on three types of data: a) Synthetic data, which is randomly generated to verify the behavior of the proposed model; b) Visual Place Categorization (VPC) 09 video dataset, which was captured in the same fashion as a real online system; c) Caltech 256 image classification dataset; and d) two UCI datasets.

For evaluation, we compare our model OSLLR-GS with the state-of-the-art methods including both online learning methods (OASIS [16], [17], [23], LMNN [18], LEGO [19], OMLLR [21] and MCML [20]) and offline (batch) training methods (K-Nearest Neighbor and Wu's method [51]). The accuracy is evaluated by a K-NN classification procedure, i.e., the testing sample is classified by a majority vote of its K neighbors, with the object being assigned to the class most common among its K nearest neighbors (K is a positive integer, e.g., 1), where the similarity is measured by the corresponding similarity learning model. The accuracy is measured by

$$Acc = \frac{\sharp \text{ of samples classified correctly}}{\sharp \text{ of all samples}}, \tag{19}$$

where $\sharp$ means the number.

For comparison, we choose two criteria as follows:

$$W_{max} = \arg\max_{W_i} Acc(i), i \in \{1, \ldots, N\} \tag{20}$$

$$\overline{W} = \sum_{i=1}^{N} \alpha^i W_i \Big/ \sum_{i=1}^{N} \alpha^i, \tag{21}$$

where N is the total number of iteration, W_i is the model matrix of ith step iteration, and $acc(i)$ is the accuracy of the corresponding W_i. The first criterion is the model with the highest accuracy and the second is the average weighted performance of the model. As the accuracy of the online similarity learning model will fluctuate in each updating iteration, the model with greater $\overline{W}$ in Eq. (21) is more robust in practice and the model in Eq. (20) is only chosen for comparison here.

5.1 Synthetic Dataset

To validate the effectiveness of the new constraints in our model in Eq. (12), i.e., low rank and group sparsity constraints, we generate the synthetic data and compare our model (OSLLR-GS) with the state-of-the-arts. Specifically, our synthetic dataset is a two-classes data with dimension as 20, and we random sample 200 samples from each class, i.e., we totally have 400 samples (30 percent training, 70 percent testing). To achieve this, we first generate two different Gaussian models with the dimension as 5, randomly sample 200 from each model, and project them into $20d$ feature space by random projection.

i. Effectiveness of the low rank constraint:
   As shown in Fig. 2, we plot the singular values of the model W ($W = XX^T$ in our case) using SVD calculation. When the data rank is 5, the rank is 3 for both our OSLLR-GS and OMLLR [21]. In comparison, the rank for OASIS [16], [17], [23] is 20. For the same size of model W, lower rank in an appropriate range means less model complexity and induces lower risk of overfitting. Therefore, Fig. 2 justifies the effectiveness of the low rank constraint.
ii. The efficiency of feature selection via group sparsity:
   The second term of Eq. (12) with the tuning parameter μ is used for feature selection with group sparsity. As shown in Fig. 3, the value of $\|X_{k.}\|$ shows the importance (weight) of the corresponding feature dimension. Our OSLLR-GS successfully selects the five most useful features out of the redundant features. In comparison, OASIS and OMLLR fail to do that.
iii. Comparison on the Convergence Rate:
   We illustrate the convergence curve of our model by adopting the synthetic dataset, where the cost is calculated by Eq. (12). We can see that the cost function of our model converges iteratively as shown in Fig. 5, where we plot the cost every 100 iterations.

In this case, we preset the feature dimension as 20 and the data rank in the embedded subspace as 5.

5.2 Visual Place Categorization 09 Dataset

The VPC 09 dataset is collected from 6 home environments using a rolling tripod plus a camera to mimic an online

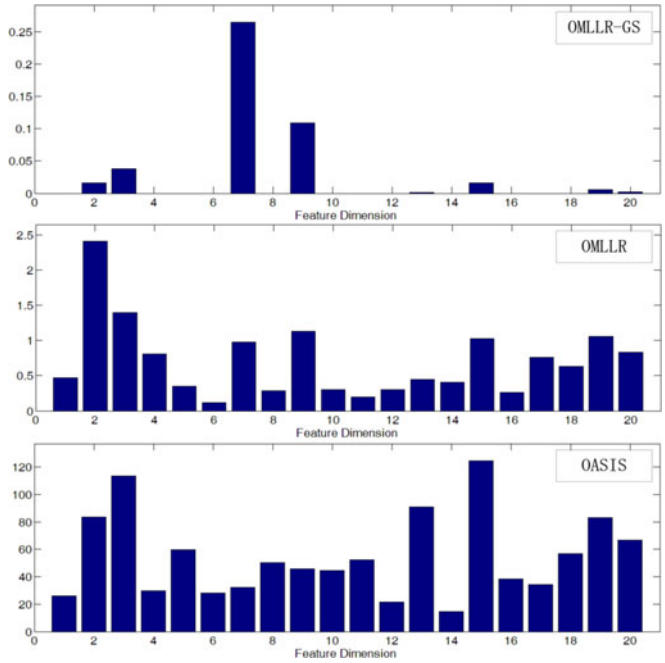


Fig. 3. Comparison of the weights for feature selection, where the results of ours, OMLLR and OASIS are shown in the first, second and last row, respectively. Ours with groups sparsity is more efficient for feature selection.

robot system, including 12 different scenarios with the resolution $1,280 \times 720$ for every image. For a fair comparison, we follow the experiments setup in [53] and also adopt 5 categories (bedroom, bathroom, kitchen, living-room, and dining-room). The CENTRIST feature [53] is extracted from each image (or frame), resulting in 1,302 dimensions with the spatial-pyramid structure. We compare our OSLLR-GS with the state-of-the-art methods, including online methods (OMLLR [21], OASIS [16], [17] and LMNN [52]) and batch training methods (K-Nearest Neighbor, 1-NN and 5-NN, and Wu's method [53]). The leave-one-out cross validation strategy is used to evaluate all algorithms. All experiments were repeated for 6 times and we report the average performance. In each run, one home was reserved for testing and all other 5 homes were combined to form a training set.

Fig. 4 compares our OSLLR-GS with OMLLR [21] and OASIS [16], [17], where each subfigure corresponds to Home 1-6 for 3 million iterations with $10k$/step. The curves of "Ours" and OMLLR are generated by weighted expectation $\overline{W}$, so convergence is guaranteed; "Ours(Iter)" and OASIS are the results of individual iterations, therefore they fluctuate step-by-step. The statistical results are shown in Table 1, our OSLLR-GS generally outperforms other online learning methods, including OMLLR, OASIS [16], [17], LMNN [52] and also K-NN based batch training methods (1-NN and 5-NN). IROS [53] is better than our model because it uses the batch training model. The accuracy of online learning models is expected to be lower than those of the batch training methods, therefore the performance of our OSLLR-GS is acceptable. For video-based frame-level scene classification, there exists a strong Markov Random property between consecutive frames. In [53], Wu et al. use

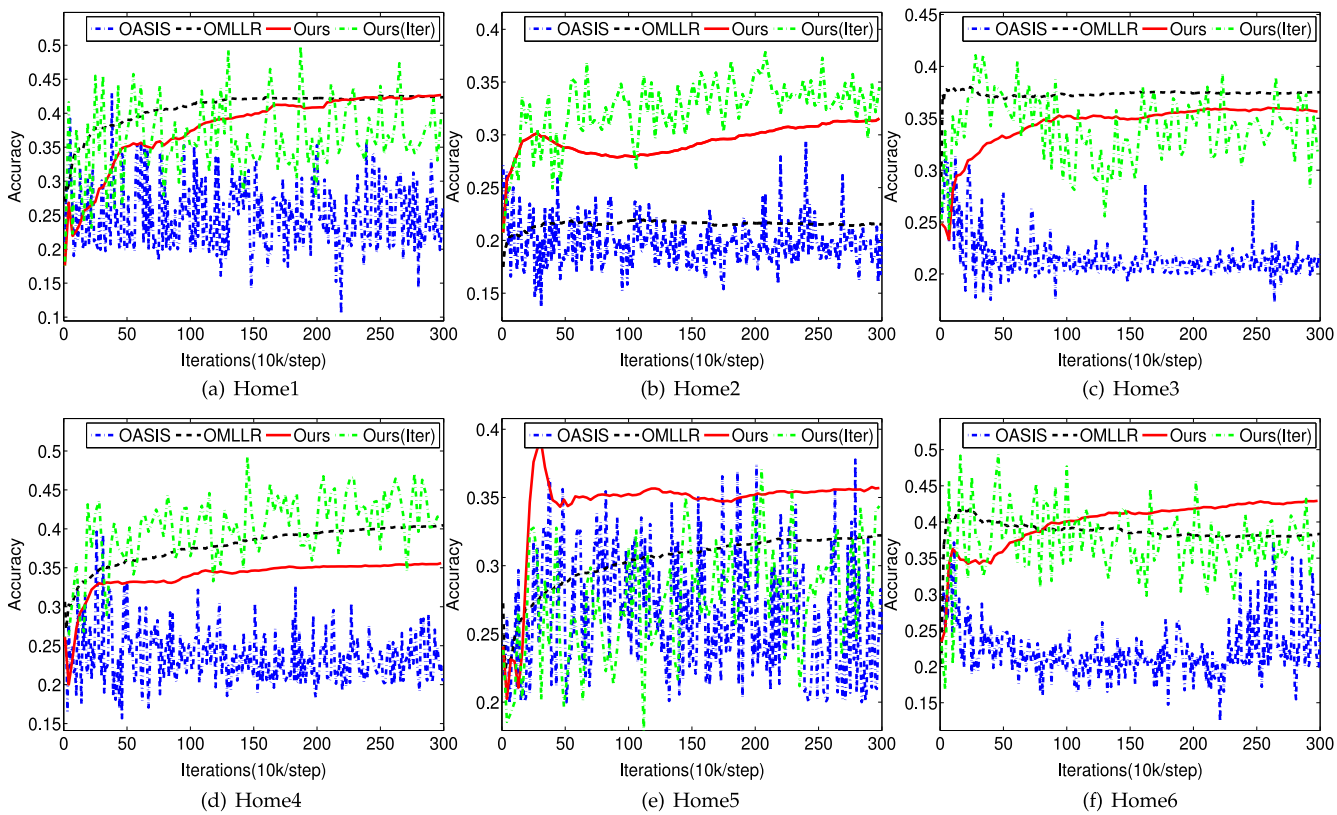


Fig. 4. The comparison of the accuracy between ours (OSLLR-GS) and other methods, such as OMLLR [21] and OASIS [16], [17] for home1-6. In each figure, the x-axis corresponds to the iteration steps (10k for each) and the y-axis is the current accuracy, where OASIS and OMLLR are denoted by dash blue and dash black line; "Ours" (sold red) is the weighted expectation result of our OSLLR-GS, and "Ours (Iter)"(dash green line) is the result of each iteration, so it fluctuate and cannot guarantee converge.

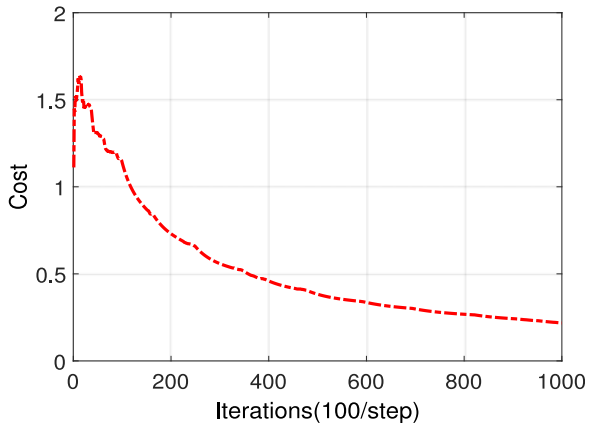


Fig. 5. Comparison of the convergence rate of our model.

temporal smoothing to improve the accuracy of the coarse result. In contrast, we only adopt a simple median filter (of width 5) for frame-level temporal smoothing. After temporal smoothing, the accuracies of both online learning and batch training are improved. Our OSLLR-GS is still better than OMLLR as shown in Table 1. According to Table 1, another interesting point is that all compared methods perform differently for different sub-datasets, i.e., Home1-6. This is because each sub-dataset is somewhat different, e.g., different scenarios, different condition.

5.3 Caltech 256 Dataset

We also test our OSLLR-GS using the Caltech 256 dataset [54]. Following [23], we select 20 and 50 classes from the Caltech 256 dataset (The class label is shown in Appendix B). For each set, images from each class were 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 image representation, we adopt the same features used in [23] with the feature dimension as 1,000 for a fair comparison. For evaluation, a standard ranking precision measure based on nearest neighbors is also used, i.e., 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, which yields a measure known as precision-at-top-k and provides 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.

Our method OSLLR-GS is compared with the state-of-the-art online similarity learning methods, including OMLLR [21], OASIS [16], [17], [23], LMNN [18], LEGO [19], MCML [20], Loreta [38], [39] and euclidean (the standard euclidean distance in feature space). The statistic results are proposed in Table 2. In general, our model OSLLR-GS outperforms other methods expect for OMLLR for Top 1 prec., including OASIS and OMLLR with $\gamma = 0$. Even the mean

TABLE 1
The Comparison of the Average Accuracy of Our OMLLR and the State-of-the-Art Methods Using VPC 09 Dataset

Filter		Methods	Home1	Home2	Home3	Home4	Home5	Home6	Avg
No	Online	Ours	42.68	31.54	36.01	35.60	35.12	42.93	37.31
		OMLLR	42.36	21.53	37.53	40.43	32.22	38.28	35.39
		OASIS [16], [17]	25.33	21.32	21.99	20.57	24.84	39.18	25.54
		LMNN [52]	39.41	28.75	36.79	39.06	30.74	34.88	34.94
No	Batch	IROS [53]	44.77	33.33	40.68	43.28	41.10	48.07	41.87
		1-NN	41.83	27.48	33.96	38.66	30.85	29.70	33.75
		5-NN	41.18	28.23	34.33	39.82	31.62	31.56	34.46
Yes	Online	Ours	45.65	34.21	38.43	39.89	38.26	44.99	40.23
		OMLLR	46.03	21.66	38.59	41.95	33.05	41.29	37.10
	Batch	IROS [53]	44.58	35.89	40.96	49.93	46.91	55.46	45.62

TABLE 2
Average Precision and Precision at Top 1, 10, and 50 of All Compared Methods

20 classes	Ours Matlab	OMLLR Matlab	OMLLR($\gamma = 0$) Matlab	OASIS Matlab	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	Euclidean -	Loreta Matlab
MeanAvg	22 ± 1.4	23 ± 1.3	21 ± 1.3	21 ± 1.4	17 ± 1.2	16 ± 1.2	14 ± 0.6	14 ± 0.7	24.3 ± 0.7
Top 1	32 ± 1.7	33 ± 1.7	29 ± 1.8	29 ± 2.6	26 ± 2.3	26 ± 2.7	26 ± 3.0	25 ± 2.6	28.4 ± 3.2
Top 10	26 ± 1.6	26 ± 1.6	23 ± 1.7	24 ± 1.9	21 ± 1.5	20 ± 1.4	19 ± 1.0	18 ± 1.0	26.4 ± 0.9
Top 50	18 ± 0.9	20 ± 1.0	17 ± 0.6	15 ± 0.4	14 ± 0.5	13 ± 0.6	11 ± 0.2	12 ± 0.2	21.4 ± 0.6
50 classes	Ours Matlab	OMLLR Matlab	OMLLR($\gamma = 0$) Matlab	OASIS Matlab	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	Euclidean -	Loreta Matlab
MeanAvg	13 ± 1.4	14 ± 0.3	13 ± 0.4	12 ± 0.4	N/A	9 ± 0.4	8 ± 0.4	9 ± 0.4	14 ± 0.6
Top 1	22 ± 1.5	22 ± 1.4	18 ± 1.5	21 ± 1.6	N/A	18 ± 0.7	18 ± 1.3	17 ± 0.9	17 ± 0.7
Top 10	15 ± 0.9	17 ± 0.3	15 ± 0.4	16 ± 0.4	N/A	13 ± 0.6	12 ± 0.5	13 ± 0.4	15 ± 0.4
Top 50	10 ± 0.5	12 ± 0.4	11 ± 0.3	10 ± 0.3	N/A	8 ± 0.3	7 ± 0.2	8 ± 0.3	12 ± 0.5

Values are averages over 5-fold cross-validations; $\pm$ values are the standard deviation across the 5 folds. OMLLR($\gamma = 0$) means no consideration of the effectiveness of low rank constraint.

TABLE 3
Effectiveness of Tuning Parameters, λ and μ, Where $\lambda \to \inf$ and $\mu = 0$ Means These Two Parameters Have No Effect

20 classes	Ours Matlab	Ours($\lambda \to \inf$) Matlab	Ours($\mu = 0$) Matlab
Mean avg prec	22 ± 1.4	21 ± 1.4	21 ± 1.5
Top 1 prec.	32 ± 1.7	29 ± 1.8	30 ± 1.6
Top 10 prec.	26 ± 1.6	23 ± 1.6	23 ± 1.7
Top 50 prec.	18 ± 0.9	18 ± 0.7	17 ± 0.5
50 classes	Ours Matlab	Ours($\lambda \to \inf$) Matlab	Ours($\mu = 0$) Matlab
Mean avg prec	13 ± 1.4	13 ± 1.0	14 ± 1.3
Top 1 prec.	22 ± 1.5	18 ± 1.4	19 ± 1.4
Top 10 prec.	15 ± 0.9	15 ± 0.8	15 ± 1.0
Top 50 prec.	10 ± 0.5	10 ± 0.4	10 ± 0.4

average precision of Loreta [38], [39] is better, its performance is not good when K is smaller, e.g., $K = 1$, which is the most important criterion due to users always adopt $K = 1$ or $K = 5$ for K-NN classification in practice.

We compare the effectiveness of tuning parameter, λ and μ, as shown in Table 3, where we set $\lambda \to \inf$ and $\mu = 0$ in order to remove the effectiveness of low rank constraint and group sparsity constraint, respectively. When neither constraint is activated, the performance of our model will deteriorate and is similar to OMLLR ($\gamma = 0$). However, our model still outperforms other state-of-the-art methods. This validates the effectiveness of these two constraints in our model.

Fig. 6 shows the precision curves for retrieval. Interestingly, when the class number increases from 20 classes to 50 classes, the performances of all methods are poor. This is because for a fixed number of training steps, e.g., 35 K iterations in our case, the higher the number of classes is, the lower the probability of different samples meeting each other is. The performance of our OSLLR-GS gets closer to the best one by OMLLR.

5.4 Comparison on the Model Parameters λ, μ and m

In this section, we compare the influence of the model parameters λ, μ and m to the model accuracy, where λ in Eq. (12) is used to control the rank redundancy, μ in Eq. (12) is to constraint the group sparsity and m is the number of columns of $X \in \mathbb{R}^{d \times m}$. Following Section 5.2, we also adopt the VPC 09 video dataset as well by using Home1 as testing set and Home 2-6 as training set. We first fix all other parameters and change the parameter λ in the range of $[0.01, 0.1, 1, 10, 100, 200, 500, 1,000]$, the results are shown in Fig. 7. We can see that the accuracy of our model fluctuates when the value of λ increasing. Intuitively, too big or too small value of λ cannot generate good result.

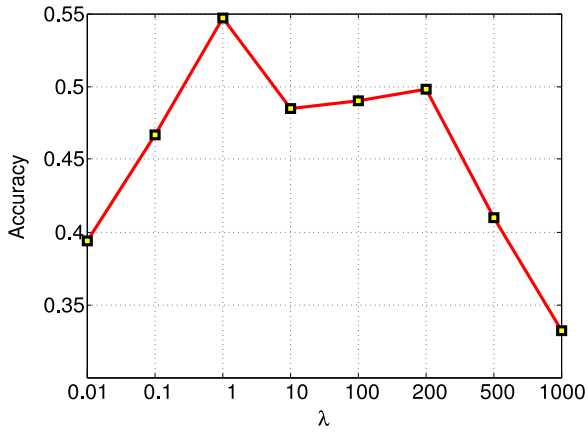


Fig. 7. Comparison of the accuracy by varying the value of λ.

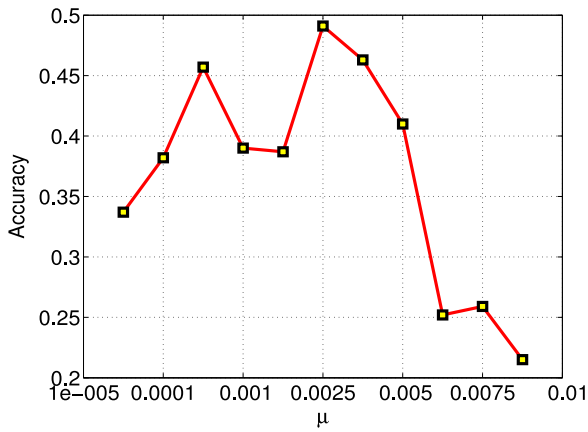
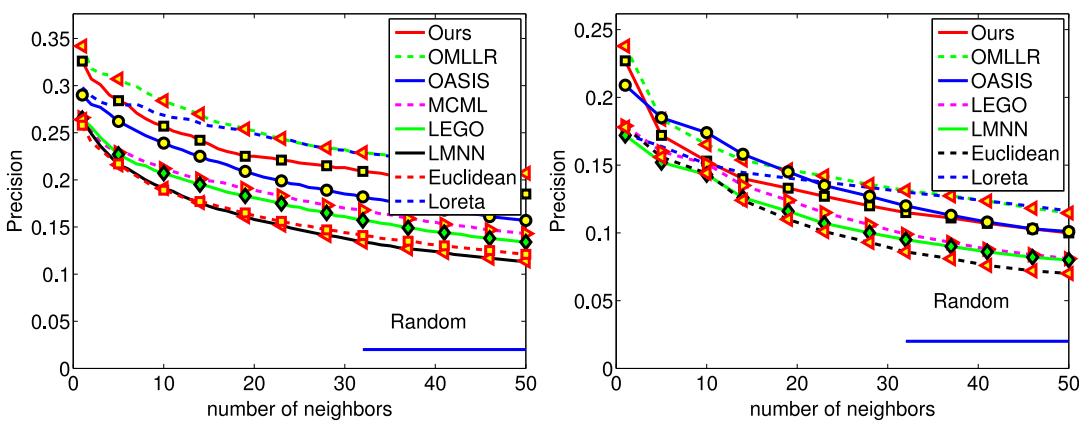


Fig. 8. Comparison of the accuracy by varying the value of μ.



(a) 20 Classes (b) 50 Classes

Fig. 6. Comparison of the performance of our model, OMLLR, OASIS, LMNN, MCML, LEGO, loreta and the euclidean metric in the 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), the result of random prediction is shown at the bottom. (a) 20 classes, (b) 50 classes.

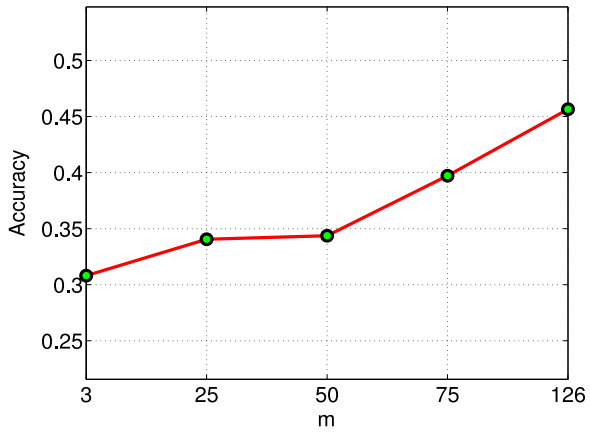


Fig. 9. Comparison of the accuracy by varying the value of m.

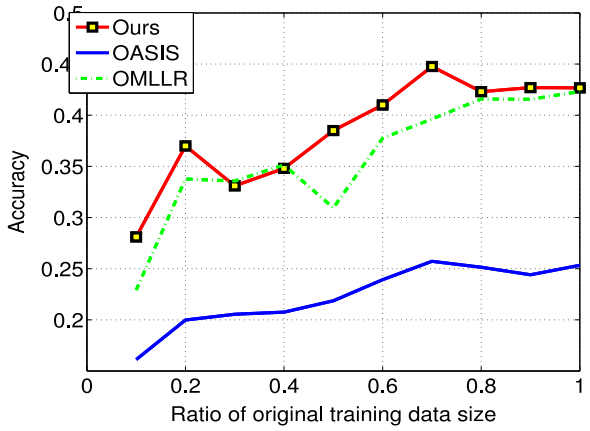


Fig. 10. Comparison the accuracy of our model with OASIS and OMLLR by varying the size of training data.

We change the parameter μ by setting μ from 10^{-5} to 0.01, and the results are shown in Fig. 8. We can see the accuracy is bad when μ is too small or too large. Especially if μ is greater than 0.1, we cannot get an acceptable result due to there are too many zero rows of the metric matrix. Actually, both λ and μ are model parameters, we should choose them properly in practice.

We then vary the parameter m by setting m as $[3, 25, 50, 75, 126]$, and the results are demonstrated in Fig. 9. It is obviously when m is small, the accuracy is bad, and the performance keep improved during the value of m increasing. This is because the larger value of m may induce a global optimal solution and the matrix X can contain enough information as well. However, in practice, a larger value of m will increase the size of the matrix X and increase both the memory and computation burden as well. Therefore, we should balance it in practice.

TABLE 5
Compare the Feature Selection Ability of Ours with the State-of-the-Arts Using UCI-Mfeat and UCI-SPECTF Dataset, Respectively

(a) UCI-Mfeat dataset							
	25	50	100	150	200	250	300
mRMR	**0.7237**	0.7658	0.8107	0.8438	0.8586	0.8556	0.8914
CSFS	0.5079	0.7805	0.8963	0.8671	0.8984	0.9103	0.8766
Ours	0.7166	**0.8739**	**0.8996**	**0.9002**	**0.9058**	**0.9105**	**0.9195**
(b) UCI-SPECTF dataset							
	5	10	15	20	25	30	
mRMR	0.6546	**0.7501**	0.7356	0.7327	**0.8052**	0.7023	
CSFS	**0.7501**	0.7155	**0.7951**	0.7531	0.7893	0.7472	
Ours	0.6372	0.7300	0.6618	**0.7718**	0.7356	**0.7515**	

5.5 Comparison on the Time Consumption

We adopt the Caltech 256 dataset to compare the time consumption of our OSLLR-GS with the state-of-the-art methods as shown in Table 4. The efficiency of our OSLLR-GS is competitived with OASIS and much faster than other methods, especially OMLLR. This is because the computational complexity of our OSLLR-GS is in the order of $O(d^2)$ with the feature dimension d and taking the columns of X as $d'(d \leq d' \leq 2d)$; in contrast, OMLLR using SVD calculation is $O(d^3)$. All the experiments are performed on the computer with 4 G RAM, Pentium IV 2.6 GHz CPU.

5.6 Comparison on the Training Data Size

In this section, we justify the effectiveness of our model by varying the size of training data. As shown in Fig. 10 adopting the Home 1 video data set, the original training data has 23,058 data and we decrease the ratio of the original training data size from 1 to 0.1 with each step of 0.1. We compare our model with both OASIS and OMLLR, ours nearly outperforms the other two methods for all cases. Moreover, we can see that the accuracy decreases dramatically when the ratio is lower than 0.5; in contrast, the accuracy is similar and competitive when the ratio is between 0.6 and 1, which justifies the effectiveness of our model to overcome overfitting.

5.7 Comparison on the Feature Selection Ability

In this section, we compare the feature selection ability of our online similarity learning model with the state-of-the-art feature selection methods, e.g., mRMR [55], CSFC [56]. Two UCI datasets are adopted here, i.e., the UCI-Mfeat

TABLE 4
Runtime (in Minutes) of All Compared Methods for Around 35 K Training Steps for 20 and 50 Classes, Respectively

	Ours Matlab	OMLLR Matlab	OASIS Matlab	OASIS	MCML Matlab+C	LEGO Matlab	LMNN Matlab+C	fastLMNN Matlab+C
20 classes	41 ± 3	550 ± 43	45 ± 8	0.15 ± 0.02	$7,425 \pm 106$	533 ± 49	631 ± 40	365 ± 62
50 classes	42 ± 5	731 ± 71	25 ± 2	1.6 ± 0.04	N/A	711 ± 28	960 ± 80	$2,109 \pm 67$

dataset,[2] which has about 2,000 instances distributed in 10 different classes with the feature dimension as 649; the UCI-SPECTF dataset,[3] which has about 267 instances distributed in two different classes with the feature dimension as 44. In order to evaluate the feature selection ability of the corresponding methods, each method is required to select the same number of feature dimensions from the original feature dimension, and then the accuracy is calculated by the traditional 1-Nearest Neighborhood (1-NN) method using the selected feature dimensions. Therefore, the greater the accuracy is, the more efficiency the feature selection ability of the corresponding method will be. For our method, the feature selection is achieved by ranking the feature dimension depending on the value of $\|\mathrm{X}_{k.}\|_2$. The statistic results are show in Table 5, where our method outperforms the other two methods in most cases especially for the UCI-Mfeat dataset with more classes and higher feature dimensions.

6 Conclusions

This paper presents a general online similarity learning framework (Online Similarity Learning via Low Rank and Group Sparsity, OMLLR-GS) to address two types of overfitting issues, i.e., the feature redundancy and the rank redundancy. For modeling, the similarity learning problem is formulated as an optimization problem. We use the max norm to restrict the metric matrix in a low rank space and the $\ell_{2,1}$ norm to pursue a sparse feature set. For optimization, we apply the stochastic proximal gradient decent method and prove a closed form updating in each iteration with the complexity of $O(d^2)$. The experiments on synthetic data have verified the ability of the proposed model to overcome the overfitting problems. Comparisons with the state-of-the-art methods on real world data have also shown that the proposed method is as efficient as the OASIS algorithm and performs as well as the OMLLR algorithm.

Appendix A

Proof to Theorem 1. The problem is convex and separable in terms of rows. We can consider each individual problem in the following:

$$\begin{aligned} &\min_x : \frac{1}{2}\|x-c\|^2 + \mu\|x\|, \\ &s.t. : \|x\| \le \lambda, \end{aligned} \tag{22}$$

where x denotes an arbitrary row of X and c is the corresponding row of C. $\|x\| \le \lambda$ means the ℓ_2 norm of arbitrary row of X is less than λ, which is equal to $\|X\|_{2,\infty} \le \lambda$ depending on the definition of the $\ell_{2,\infty}$ norm of X. Since this is a strongly convex problem, the optimal solution is unique and satisfies the optimality condition

$$x^* - c + \mu\partial\|x^*\| \cap \mathcal{N}_{\|x\|\le\lambda}(x^*) \neq \emptyset,$$

where $\partial\|x\|$ is the sub-differential set of $\|x\|$ and $\mathcal{N}_{\|x\|\le\lambda}(x^*)$ is the normal cone on x^* of the set $\|x\| \le \lambda$. It suffices to verify that $x^* = \min(\frac{\lambda}{\|c\|}, \max(1-\frac{\mu}{\|c\|}, 0))c$ satisfies the optimality condition. First we have

$$\partial\|x\| = \begin{cases} \{\frac{x}{\|x\|}\} & \text{if } x \neq 0 \\ \{x : \|x\| \le 1\} & \text{if } x = 0 \end{cases}, \tag{23}$$

and

$$\mathcal{N}_{\|x\|\le\lambda}(x) = \begin{cases} \{0\} & \text{if} \|x\| < \lambda \\ \{tx : t \ge 1\} & \text{if} \|x\| = \lambda \\ \emptyset & \text{otherwise.} \end{cases} \tag{24}$$

One can verify that x^* satisfies the optimality condition by enumerating the following three situations

i $\|c\| \le \mu$: $x = 0$;
ii $\lambda + \mu \ge \|c\| > \mu$: $x = (1 - \mu/\|c\|)c$;
iii $\lambda + \mu < \|c\|$: $x = \lambda c/\|c\|$.

It completes the proof. □

Appendix B

For the Caltech 256 dataset, we adopt 20 classes and 50 classes for evaluation, which are defined the same as [21], [23]. The labels are defined as below:

i. 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.
ii. 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, picnictable, elk, kangaroo-101, mattress, toaster, electric-guitar-101, bathtub, gorilla, jesus-christ, cormorant, mandolin, lighthouse, cake, tricycle, speed-boat, computer-mouse, superman, chimp, pram, friedegg, fighter-jet, unicorn, greyhound, grasshopper, goose, iguana, drinking-straw, snake, hotdog.

Acknowledgments

Y. Cong and J. Liu are co-first authors and contribute the same to this paper. This work is supported by NSFC (61375014, U1613214, 61533015), CAS-Youth Innovation Promotion Association Scholarship (2012163) and also the foundation of Chinese Scholarship Council.

References

[1] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Michigan State Universiy, vol. 2, no. 2, 2006.
[2] L. Yang, "An overview of distance metric learning," in *Proc. Comput. Vis. Pattern Recog. Conf.*, 2007, pp. 1–8.
[3] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learning*, vol. 5, no. 4, pp. 287–364, 2012.
[4] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," Tech. Rep., pp. 1–59, 2013.
[5] M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann, "Stationarity of matrix relevance LVQ," in *Proc. Int. Joint Conf. Neural Netw.*, 2015, pp. 1–8.
[6] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, "Robust distance metric learning with auxiliary knowledge," in *Proc. 21st Int. Jont Conf. Artif. Intell.*, 2009, pp. 1327–1332.

2. http://archive.ics.uci.edu/ml/datasets/Multiple+Features
3. http://archive.ics.uci.edu/ml/datasets/SPECTF+Heart

[7] Y. Yang, F. Shen, H. Shen, H. Li, and X. Li, "Robust discrete spectral hashing for large-scale image semantic indexing," *IEEE Trans. Big Data*, vol. 1, no. 4, pp. 162–171, Oct.-Dec. 2015.

[8] X. Tian, Y. Lu, N. Stender, L. Yang, and D. Tao, "Exploration of image search results quality assessment," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 95–108, Jul.-Sep. 2015.

[9] W. Shao, F. D. Salim, A. Song, and A. Bouguettaya, "Clustering big spatiotemporal-interval data," *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 190–203, Jul.-Sep. 2016.

[10] D. S. Hochbaum and P. Baumann, "Sparse computation for large-scale data mining," *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 151–174, Apr.-Jun. 2016.

[11] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learning*, 2007, pp. 209–216.

[12] A. Frome, Y. Singer, and J. Malik, "Image retrieval and classification using local distance functions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, vol. 19, Art. no. 417.

[13] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 16, 2004, Art. no. 41.

[14] B. Liu, M. Wang, R. Hong, Z. Zha, and X.-S. Hua, "Joint learning of labels and distance metric," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 40, no. 3, pp. 973–978, Jun. 2010.

[15] D. Tao, L. Jin, Y. Wang, and X. Li, "Person reidentification by minimum classification error-based kiss metric learning," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 242–252, Feb. 2015.

[16] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "An online algorithm for large scale image similarity learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, vol. 21, pp. 306–314.

[17] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learning Res.*, vol. 7, 2006, Art. no. 585.

[18] K. Weinberger and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learning Res.*, vol. 10, pp. 207–244, 2009.

[19] P. Jain, B. Kulis, I. Dhillon, and K. Grauman, "Online metric learning and fast similarity search," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 761–768.

[20] A. Globerson and S. Roweis, "Metric learning by collapsing classes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 451–458.

[21] Y. Cong, J. Liu, J. Yuan, and J. Luo, "Self-supervised online metric learning with low rank constraint for scene categorization," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3179–3191, Aug. 2013.

[22] J. Lee, B. Recht, R. Salakhutdinov, and N. Srebro, "Practical large-scale optimization for max-norm regularization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1297–1305.

[23] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learning Res.*, vol. 11, pp. 1109–1135, 2010.

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1701–1708.

[25] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for web image search," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4649–4661, Nov. 2012.

[26] M. Wang, D. Ji, Q. Tian, and X.-S. Hua, "Intelligent photo clustering with user interaction and distance metric learning," *Pattern Recog. Lett.*, vol. 33, no. 4, pp. 462–470, 2012.

[27] J. Mei, M. Liu, H. R. Karimi, and H. Gao, "LogDet divergence-based metric learning with triplet constraints and its applications," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4920–4931, Nov. 2014.

[28] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[29] C. Xu, D. Tao, and C. Xu, "Large-margin multi-view information bottleneck," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1559–1572, Aug. 2014.

[30] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 227–241, 2017.

[31] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Sci.*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[32] Y. Ying, K. Huang, and C. Campbell, "Sparse metric learning via smooth optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2214–2222.

[33] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, vol. 22, pp. 862–870.

[34] W. Bian and D. Tao, "Constrained empirical risk minimization framework for distance metric learning," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 8, pp. 1194–1205, Aug. 2012.

[35] G. Kunapuli and J. Shavlik, "Mirror descent for metric learning: A unified approach," in *Proc. Eur. Conf. Mach. Learning*, 2012, pp. 859–874.

[36] K. Huang, Y. Ying, and C. Campbell, "GSML: A unified framework for sparse metric learning," in *Proc. 9th IEEE Int. Conf. Data Mining*, 2009, pp. 189–198.

[37] G. Meyer, S. Bonnabel, and R. Sepulchre, "Regression on fixed-rank positive semidefinite matrices: A Riemannian approach," *J. Mach. Learning Res.*, vol. 12, pp. 593–625, 2011.

[38] U. Shalit, D. Weinshall, and G. Chechik, "Online learning in the manifold of low-rank matrices," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 2128–2136.

[39] U. Shalit, D. Weinshall, and G. Chechik, "Online learning in the embedded manifold of low-rank matrices," *J. Mach. Learning Res.*, vol. 13, no. 1, pp. 429–458, 2012.

[40] W. Liu, C. Mu, R. Ji, S. Ma, J. R. Smith, and S.-F. Chang, "Low-rank similarity metric learning in high dimensions," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2792–2799.

[41] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning," in *Proc. Int. Conf. Mach. Learning*, 2013, pp. 615–623.

[42] J. Weston, S. Bengio, and N. Usunier, "WSABIE: Scaling up to large vocabulary image annotation," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, vol. 11, pp. 2764–2770.

[43] R. Foygel, N. Srebro, and R. Salakhutdinov, "Matrix reconstruction with the local max norm," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 944–952.

[44] A. Jalali and N. Srebro, "Clustering using max-norm constrained optimization," Tech. Rep., pp. 1–20, 2012.

[45] G. J. O. Jameson, "Summing and nuclear norms in Banach space theory," in *London Mathematical Society Student Texts*. Cambridge, U.K.: Cambridge Univ. Press, 1987.

[46] S. Burer and R. D. C. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Math. Program.*, vol. 95, pp. 329–357, 2003.

[47] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Math. Program.*, vol. 129, no. 2, pp. 163–195, 2011.

[48] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[49] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2719–2727.

[50] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Math. Program.*, vol. 155, no. 1–2, pp. 267–305, 2016.

[51] J. Wu, H. Christensen, and J. Rehg, "Visual place categorization: Problem, dataset, and algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4763–4770.

[52] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learning Res.*, vol. 10, pp. 207–244, Feb. 2009.

[53] J. Wu and J. Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1489–1501, Aug. 2011.

[54] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep., California Institute of Technology, pp. 1–20, 2007.

[55] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[56] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1171–1177.

Yang Cong (S'09-M'11-SM'15) received the BSc degree from Northeast University, in 2004 and the PhD degree from State Key Laboratory of Robotics, Chinese Academy of Sciences, in 2009. He is a full professor of Chinese Academy of Sciences. He was a research fellow of National University of Singapore (NUS) and Nanyang Technological University (NTU) from 2009 to 2011, respectively; and a visiting scholar with the University of Rochester. He has served on the editorial board of the *Journal of Multimedia*. His current research interests include image processing, compute vision, machine learning, multimedia, medical imaging, data mining, and robot navigation. He has authored more than 60 technical papers. He is a senior member of the IEEE.

Ji Liu received the BS degree in automation from the University of Science and Technology of China, in 2005, the MS degree in computer science from Arizona State University, in 2010, and the PhD degree in computer sciences from the University of Wisconsin-Madison, in 2014. He is currently an assistant professor in the Computer Science Department and Data Science Institute, University of Rochester. His research interests include machine learning and optimization and their applications in big data analytics, data mining, computer vision, medical data analysis, etc. He won the KDD best research paper award honorable mention in 2010.

Baojie Fan received the BS and MS degrees in automation from Qufu Normal University and Northwest University China, in 2006 and 2008, respectively, and the PhD degree in pattern recognition and intelligent system from the State Key Laboratory of Robotics, Shenyang Institute Automation, Chinese Academy of Sciences. His major research interests include UAV vision system, space robot, object tracking, and pattern recognition.

Peng Zeng received the BE degree in computer science from Shandong University, in 1998 and the PhD degree from SIA, in 2005. He is currently a professor and PhD supervisor in the Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS). He is also the director of the Department of Industrial Control Network and System of SIA, CAS. His research interests include industrial communication, smart grids, demand response, and wireless sensor networks. He is an expert member of the IEC TC65 WG16, a member of the standards committee of SP100, and a member of the Wireless WG of FieldBus Foundation.

Haibin Yu is a professor of Chinese Academy of Sciences. His research interests include smart grid, wireless sensor networking, control theory, and large-scale social analysis. He is the deputy director of the 10th Council of Chinese Association of Automation (CAA), Member of the Expert Group in Intelligent Manufacturing Special Program, the Ministry of Science and Technology of P. R. China, directing member at the China National Technical Committee for Industrial Process Measurement Control and Automation Standardization (SAC/TC124), and deputy directing member at the China National Technical Committee for Industrial Automation System and Integration Standardization (SAC/TC159). He has actively served on multiple technical committees. He is also a fellow of the ISA.

Jiebo Luo (S93-M96-SM99-F09) joined the Department of Computer Science, University of Rochester, in 2011 after a prolific career of more than 15 years with Kodak Research. His research spans computer vision, machine learning, data mining, social media, and biomedical informatics. He has authored more than 300 technical papers and more than 90 US patents. He has served as the program chair of ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, as well as on the editorial boards of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Trans. on Multimedia*, *IEEE Trans. on Circuits and Systems for Video Technology*, the *Pattern Recognition*, the *Machine Vision and Applications*, and the *ACM Trans. on Intelligent Systems and Technology*. He is a fellow of the SPIE, the IEEE, and the IAPR.