Combinative hypergraph learning for semi-supervised image classification

Binghui Wei [a,b], Ming Cheng [a,*], Cheng Wang [a], Jonathan Li [a,c]

[a] Computer Science Department, School of Information Science and Engineering, Xiamen University, Xiamen, 361005, China
[b] College of Applied Science, Jiangxi University of Science and Technology, Ganzhou, 341000, China
[c] Faculty of Environment, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

ARTICLE INFO

ABSTRACT

Recent years have witnessed a surge of interest in hypergraph-based transductive image classification. Hypergraph-based transductive learning models the high-order relationship of samples by using a hyperedge to link multiple samples. In order to extend the high-order relationship of samples, we incorporate linear correlation of sparse representation to hypergraph learning framework to improve learning performance. In this paper, we present a new transductive learning method called combinative hypergraph learning (CHL). CHL captures the similarity between two samples in the same category by adding sparse hypergraph learning to conventional hypergraph learning. And more, we propose two strategies to combine the two hypergraph learning methods. Experimental results on two image datasets have demonstrated the effectiveness of CHL in comparison to the state-of-the-art methods and shown that our proposed method is promising.

1. Introduction

Currently, there is widespread interest in the development of image classification using transductive learning. Because transductive learning explores not only labeled data but also unlabeled data, it achieves a performance better than the methods that learn about classifiers based only on labeled data. Its success is based on one of the following two assumptions: cluster and manifold assumptions. The cluster assumption supposes that the decision boundary should not cross high-density regions, whereas the manifold assumption means that each class lies on an independent manifold. Therefore, more and more researchers have been devoted to transductive learning based on these assumptions in recent years [1–36]. Rosenberg et al. proposed self-training to train object detection systems [8]. Blum and Mitchell used co-training to classify web pages and provided empirical results which achieved significant improvement of hypotheses on real web-page data in practice [9]. Joachims introduced transductive support vector machines (TSVMs) for text classification [10]. Another important transductive learning method is graph-based learning, which is the derived form of hypergraph-based learning on which we have focused in this paper.

The graph-based learning [15–41] achieves a promising performance between the existing transductive learning methods. Its development goes through two stages: simple-graph learning and hypergraph learning. This type of learning is built on a graph, in which vertices are samples and edge weights indicate the similarity between two samples. However, the simple-graph learning methods consider only the pairwise relationship between two samples, and they ignore the relationship in a higher order. Hypergraph learning aims to get the relationship between several samples in a higher order. Unlike a simple graph that has an edge between two vertices, a set of vertices is connected by a hyperedge in a hypergraph, and each hyperedge is assigned a weight. Hypergraph learning derives from simple-graph learning, and thus it achieves a promising performance in many applications. For example, Agarwal et al. utilized hypergraph to clustering by using clique average [14]. Zass and Shashua adopted hypergraph in image matching by using convex optimization [15]. Sun et al. utilized hypergraph to problems of multi-label learning [16]. Huang et al. applied hypergraph cut to video segmentation [17]. Tian et al. proposed hypergraph-based learning algorithm to classify gene expression data by using biological knowledge as a constraint [18]. Huang et al. formulated the task of image clustering as a problem of hypergraph partition [52]. A hypergraph-based image retrieval approach is proposed in [19]. This approach constructs a hypergraph by generating a hyperedge from each sample and its neighbors, and hypergraph-based ranking is then performed. Wong and Lu proposed hypergraph-based 3-D

* Corresponding author.
  E-mail address: chm99@xmu.edu.cn (M. Cheng).

object recognition [20]. Bu et al. [21] developed music recommendation by modeling the relationship of different entities through a hypergraph to include music, tag, and users.

In hypergraph learning, the weights of the hyperedges are empirically set according to certain rules. Zhou et al. connected k neighbor points of a given point as a hyperedge [37]. Yu et al. proposed a hypergraph learning method by adaptively coordinating the weights of the hyperedge [38]. In essence, they chose multiple neighborhoods to construct multiple hyperedges for a given point. By this way, how to define a hyperedge and the weight of a hyperedge is the succeeding problem we should take into account. There are mainly two methods to select the neighborhood of a given point, one is to define a hyperedge with fixed number neighbor points, and the other one is to define a ball whose radius is less than some threshold. However, these hypergraph-based methods focus only on proximity relation for distance. There is some other high-order relationship, such as linear relationship. For example, given two vectors a and b for existing $a = k^*b$, the distance between a and b may be very large, but, they are similar for one factor difference based on linear relationship. Therefore, the linear relationship is another one we should take into account.

This paper takes into account the clustering assumption that the similar points in feature space more likely belong to the same category. Then, we define the similarity by two assumptions that we call as distance-similarity and linear-similarity. Distance-similarity is that the points derived from a category are located close to each other. The neighborhood size of the hypergraph based on this assumption is chosen as a fixed number. Linear-similarity is that a data point can more likely be represented linearly by the data points nearby which belong to the same category as this data point. It is analogous to manifold assumptions aforementioned. The neighborhood size of the hypergraph based on this assumption is unfixed since it is decided by the sparse representation method. Inspired by the two assumptions, we construct the following two kinds of hypergraphs on a data set, one is conventional hypergraph which is based on the distance-similarity, the other one is sparse hypergraph which is based on linear-similarity and derived from the thinking in Refs. [53,54]. We linearly combine the two hypergraph learning methods by two strategies. The first strategy is to combine the hyperedges of the two hypergraphs to form a new hyperedge set. The second strategy is to linearly combine the confidence of the labeling of the two hypergraph learning methods to form a new confidence of the labeling to define the label of data points.

The contributions of this paper are as follows:

1) A novel algorithm named Combinative Hypergraph Learning (CHL) is proposed for image classification. To consider the high-order information, we incorporate sparse representation into the standard hypergraph learning framework. Hence, our algorithm achieves a better performance than that with only conventional hypergraph learning or with only sparse hypergraph learning.
2) We provide two strategies to combine linearly conventional hypergraph learning and sparse hypergraph learning. The essence is that we combine linearly the results of the two hypergraph learning methods with same weights for one strategy and with different weights for the other strategy.
3) We conduct comprehensive experiments to empirically analyze our algorithm on two image databases. The experimental results demonstrate that our algorithm outperforms other methods including Transductive SVM [10], Simple Graph-Based Learning [50], and Semi-supervised Discriminant Analysis (SDA) [42,51] classifications.

The rest of this paper is organized as follows. Section II describes the proposed image classification by combinative hypergraph learning. Section III shows experiments on practical image datasets. Section IV concludes the paper.

2. Hypergraph learning

This section introduces conventional hypergraph learning and sparse hypergraph learning first, and shows combinative hypergraph learning theory with two strategies in the following text. In Table 1, we provide important notations used in the rest of this paper to present the technique details of the proposed method. Fig. 1 illustrates the whole framework of our method.

2.1. Conventional hypergraph learning

Given c categories of images including m training data points (x_1, y_1), …, (x_m, y_m), and n testing data points (x_{m+1}, $\mathbf{0}$), …, (x_{m+n}, $\mathbf{0}$), where $x_i \in \mathbb{R}^d$, $1 \le i \le m+n$ is sampled from the input space; $y_i = [0, …, 1, …, 0]^T \in \mathbb{R}^c$, $1 \le i \le m$, is the label vector of x_i, where the g-th component is 1 if x_i belongs to the g-th category, otherwise, 0; and $\mathbf{0}$ is a vector with c components of zero.

Table 1
Important notations used in this paper.

| Notations	Descriptions
$\mathbf{H} = (\mathbf{x}, \varepsilon)$	The representation of a hypergraph, where $\mathbf{x}$ and ε indicate the sets of vertices and hyperedges, respectively
$\mathbf{A}$	The incidence matrix for the hypergraph
dist	The distance between two samples
φ	The diagonal matrix of the vertex degrees
γ	The diagonal matrix of the hyperedge degrees
ω	The diagonal weight matrix and its (i,i)-th element is the weight of the i-th hyperedge
$\mathbf{L}$	The constructed hypergraph Laplacian matrix
y_i	The label vector for i-th class. Its j-th element is 1 if the j-th object belongs to the i-th class, and otherwise it is 0.
$\mathbf{F} = [f_1, f_2, …, f_c]$	$\mathbf{F}$ represents the relevance score matrix for all samples, and f_i is the to-be-learnt relevance score vector for class i.
c	The number of classes in the dataset
m	The number of labeled images in the dataset
n	The number of images in the dataset
l	The number of hyperedges
$\mathbf{A}^{sp}$, dist^{sp}, $\mathbf{F}^{sp}$	The superscript "sp" denotes sparse-based which show difference to $\mathbf{A}$, dist and $\mathbf{F}$, respectively
$\mathbf{A}^*$, $\mathbf{F}^*$	The superscript "*" denotes the concatenated results.
$\Sigma = [\varsigma_1, \varsigma_2, …\varsigma_d]$	The base of sparse representation
$w = [w_1, w_2, …w_d]^T$	The coefficient vector of sparse representation
$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, …\mathbf{w}_n]$	The coefficient matrix of sparse representation

a

b
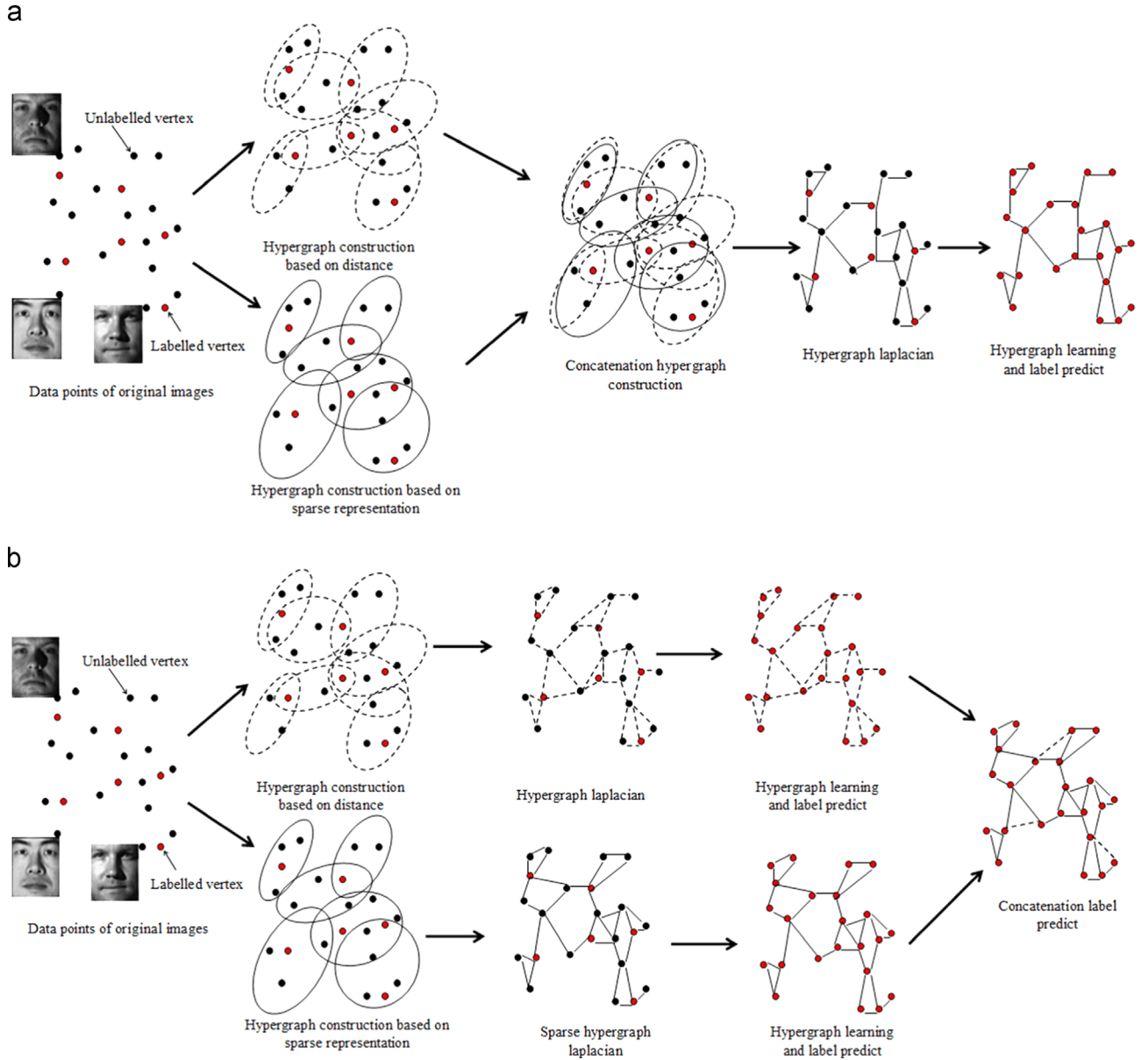
Fig. 1. Workflow of the hypergraph learning for image classification with two combinative strategies. (a) Strategy one and (b) strategy two.

The hypergraph, $\mathbf{H} = (\mathbf{x}, \varepsilon)$, is formed by the vertex set, $\mathbf{x}$, and the hyperedge set, ε, an incidence matrix, $\mathbf{A}$, whose size is $|\mathbf{x}| \times |\varepsilon|$, denotes the hypergraph with the following elements:

$$\mathbf{A}(i,j) = \begin{cases} 1, & \text{if} \quad \mathbf{x}_i \in \varepsilon_j \\ 0, & \text{if} \quad \mathbf{x}_i \notin \varepsilon_j \end{cases}, \tag{1}$$

where ε_j denotes the j-th element of the hyperedge set. The distance between two data points is

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right)$$

$$\sigma = \sqrt{\frac{1}{m+n-1}\sum_{\delta=1}^{m+n}\|\mathbf{x}_\delta - \overline{\mathbf{x}}\|^2} \quad \overline{\mathbf{x}} = \frac{1}{m+n}\sum_{\delta=1}^{m+n}\mathbf{x}_\delta. \tag{2}$$

We use $\boldsymbol{\varphi}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ to denote diagonal matrices of vertex degrees, hyperedge degrees and hyperedge weights, respectively. φ_i denotes the entry (i, i) of matrix $\boldsymbol{\varphi}$, ω_i and γ_i have similar

meanings. Then, the initial weight, ω_i, is

$$\omega_i = \sum_{\mathbf{x}_j \in \varepsilon_i} \text{dist}(\mathbf{x}_i, \mathbf{x}_j). \tag{3}$$

Based on $\mathbf{A}$, the i-th vertex degree, φ_i, is

$$\varphi_i = \sum_{\varepsilon_j \in \varepsilon} \omega_j \mathbf{A}(i,j), \tag{4}$$

and the i-th hyperedge degree, γ_i, is

$$\gamma_i = \sum_{\mathbf{x}_j \in \mathbf{x}} \mathbf{A}(j,i). \tag{5}$$

In this letter, we adopt the regularization framework proposed in Yu et al. [38], i.e.,

$$\arg\min_{\mathbf{F},\boldsymbol{\omega}} \sum_{\eta=1}^{c} (\mathbf{f}_\eta^{\mathrm{T}} \mathbf{L} \mathbf{f}_\eta) + \lambda \|\mathbf{f}_\eta - \mathbf{Y}_\eta\|^2 + \mu \|\text{diag}(\boldsymbol{\omega})\|^2$$

$$\text{s.t.} \sum_{j=1}^{l} \omega_j = 1, \quad \omega_j \geq 0, \quad j = 1, \ldots, l, \tag{6}$$

where $\mathbf{L} = \mathbf{I} - \boldsymbol{\varphi}^{-(1/2)} \mathbf{A} \boldsymbol{\gamma}^{-1} \boldsymbol{\omega} \mathbf{A}^{\mathrm{T}} \boldsymbol{\varphi}^{-(1/2)}$; l is the number of hyperedges; diag($\boldsymbol{\omega}$) is the diagonal vector of $\boldsymbol{\omega}$, i.e., ($\omega_1, \omega_2, ..., \omega_l$); c is the number of classes; $\mathbf{F}$ is a matrix, $\mathbf{F} = (\boldsymbol{f}_1, ..., \boldsymbol{f}_c) \in \mathfrak{R}^{(m+n) \times c}$, $\boldsymbol{f}_\eta$ is the confidence of the labeling for the data belonging to the ηth category; $\lambda > 0$ and $\mu > 0$ are two trade-off parameters to balance the empirical loss, the weight and the regularization; $\mathbf{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_m, \mathbf{0}, ..., \mathbf{0})^{\mathrm{T}}$, $\mathbf{Y} \in \mathfrak{R}^{(m+n) \times c}$, $\mathbf{Y}_\eta$ is the ηth column of $\mathbf{Y}$. The third term is introduced to avoid a degenerate solution caused by the former two terms existing only in the regularization. Here we add two constraints, one to fix the summation of the weights $\sum_{j=1}^{l} \omega_j = 1$ and the other to avoid negative weight $\omega_j \geq 0$.

Because the classifier function is not jointly convex with respect to $\mathbf{F}$ and $\boldsymbol{\omega}$, we solve one variable by fixing another variable.

First we initialize $\boldsymbol{\omega}$ with (3), so the solution of $\mathbf{F}$ becomes

$$\mathbf{F} = \frac{\lambda}{\lambda+1} \left(\mathbf{I} - \frac{\boldsymbol{\varphi}^{-(1/2)} \mathbf{A} \boldsymbol{\gamma}^{-1} \boldsymbol{\omega} \mathbf{A}^{\mathrm{T}} \boldsymbol{\varphi}^{-(1/2)}}{1+\lambda} \right)^{-1} \mathbf{Y}. \tag{7}$$

Then we update the weights, $\boldsymbol{\omega}$, with an iterative coordinate descent method as follows:

$$\begin{cases} \omega_i^{t+1} = 0, \omega_j^{t+1} = \omega_i^t + \omega_j^t, & \text{if } 2\mu(\omega_i^t + \omega_j^t) + (s_j - s_i) \leq 0 \\ \omega_i^{t+1} = \omega_i^t + \omega_j^t, \omega_j^{t+1} = 0, & \text{if } 2\mu(\omega_i^t + \omega_j^t) + (s_i - s_j) \leq 0 \\ \omega_i^{t+1} = \frac{2\mu(\omega_i^t + \omega_j^t) + (s_j - s_i)}{4\mu}, & \omega_j^{t+1} = \omega_i^t + \omega_j^t - \omega_i^{t+1}, \quad \text{otherwise} \end{cases}, \tag{8}$$

where $s_i = \sum_{z=1}^{c} -(r_i^z)^2 (\gamma_i)^{-1}$, $\boldsymbol{r}^z = \boldsymbol{f}_z^{\mathrm{T}} \boldsymbol{\varphi}^{-1/2} \mathbf{A}$, r_i^z is the i-th component of $\boldsymbol{r}^z$; ω_i^t denotes the t-th iteration of ω_i and the initial iteration t is 0. Based on the coordinate descent method, an iterative process alternately updates the labels and the weights.

In the next iteration, we calculate the new $\mathbf{F}$ with the new $\boldsymbol{\omega}$. The iteration ceases at a given state. After obtaining $\mathbf{F}$, we set the g-th class to the i-th data point if the g-th component is the maximum in the i-th row of $\mathbf{F}$. A more detailed solution of (6) appears in Ref. [38].

2.2. Sparse hypergraph learning

Sparse hypergraph learning is similar to conventional hypergraph learning. Therefore, this section will show only the difference between sparse hypergraph learning and spectral hypergraph learning to avoid repetition. It is the hyperedge construction and the hyperedge weight definition which will be introduced after the introduction of sparse representation in this section.

The application of sparse representation to computer vision has attracted a lot of attention in recent years [43–46]. This paper solves the sparse representation problem based on ℓ_1 minimization.

Given a vector $\boldsymbol{x}$ in R^d, which can be represented on the basis of d vectors $\{\varsigma_i \in \mathrm{R}^n\}_{i=1}^d$. By setting a matrix $\boldsymbol{\Sigma} = [\varsigma_1, \varsigma_2, \cdots, \varsigma_d]$ we can rewrite $\boldsymbol{x}$ as

$$\boldsymbol{x} = \sum_{i=1}^{d} w_i \varsigma_i = \boldsymbol{\Sigma} \boldsymbol{w} \tag{9}$$

where $\boldsymbol{w} = [w_1, w_2, ..., w_d]^T$. Both $\boldsymbol{x}$ and $\boldsymbol{w}$ represent the same data point, one in the space domain and the other in the $\boldsymbol{\Sigma}$ domain. Our object is to find a sparse representation of $\boldsymbol{x}$ in a properly chosen basis $\boldsymbol{\Sigma}$, namely, $\boldsymbol{w}$ must have as few nonzero components as possible. According to [48], such a sparse representation can be obtained by solving the optimization problem

$$\min \|\boldsymbol{w}\|_0 \quad \text{subject to } \boldsymbol{x} = \boldsymbol{\Sigma} \boldsymbol{w}, \tag{10}$$

where $\|\boldsymbol{w}\|_0$ is the ℓ_0 norm of $\boldsymbol{w}$, i.e., the number of nonzero entries. However, such an optimization problem is in general non-convex and NP-hard. According to Refs. [47,48], we can replace the non-convex optimization in (10) by the following convex ℓ_1

optimization formulation:

$$\min \|\boldsymbol{w}\|_1 \quad \text{subject to } \boldsymbol{x} = \boldsymbol{\Sigma} \boldsymbol{w}. \tag{11}$$

Now, consider all data points in a dataset, $\mathbf{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n]$. Each data point, $\boldsymbol{x}_i$, has a sparse representation $\boldsymbol{w}_i$. By setting the i-th element of $\boldsymbol{w}_i$, $(\boldsymbol{w}_i)_i = 0$, the optimization formulation can be rewritten as

$$\min \sum_{i=1}^{n} \|\boldsymbol{w}_i\|_1 \quad \text{subject to } \boldsymbol{x}_i = \mathbf{x}^T \boldsymbol{w}_i. \tag{12}$$

Assume that the data set $\mathbf{x}$ is drawn from a union of c independent linear subspaces, namely, that the data set include c categories of the object. According to Ref. [54], we can obtain block sparse solutions with the nonzero block corresponding to points in the same subspace if the aforementioned assumption holds. We can recover a block sparse representation of a new data point as a linear combination of the points in the same subspace. This means that a data point $\boldsymbol{x}_i$ and the data points whose index corresponds to nonzero entry of $\boldsymbol{w}_i$ are derived from the same category.

We now show how to define a hyperedge and its weight, which are different from spectral hypergraph construction. For discrimination, we use $\mathbf{A}^{sp}$, dist^{sp} and $\mathbf{F}^{sp}$ to replace the incidence matrix $\mathbf{A}$, distance between two data points dist and classifier function $\mathbf{F}$ shown in the above section, respectively. We define $\mathbf{A}^{sp}$ as

$$\mathbf{A}^{sp}(i,j) = \begin{cases} 1, & \text{if } i = j \\ 1, & \text{if } |(\boldsymbol{w}_i)_j| > 0, \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $|(\boldsymbol{w}_i)_j|$ denotes the absolute of j-th entry of $\boldsymbol{w}_i$, and dist^{sp} as

$$\text{dist}^{sp}(\boldsymbol{x}_i, \boldsymbol{x}_j) = |(\boldsymbol{w}_i)_j|. \tag{14}$$

The following processing is the same as conventional hypergraph learning. Here, we take $\mathbf{F}^{sp}$ as the solution of the sparse hypergraph learning.

2.3. Combinative hypergraph learning

From the above two learning methods, we obtain two hypergraphs and combine them with two strategies.

2.3.1. Strategy one

Assume that we obtain two incidence matrixes $\mathbf{A}$ and $\mathbf{A}^{sp}$ of the hypergraph aforementioned. We concatenate $\mathbf{A}$ and $\mathbf{A}^{sp}$ to a new incidence matrix $\mathbf{A}^*$ whose columns number is the sum columns of $\mathbf{A}$ and $\mathbf{A}^{sp}$ as follows:

$$\mathbf{A}^* = [\mathbf{A} \mathbf{A}^{sp}], \tag{15}$$

where the rows number of $\mathbf{A}$ is equal to that of $\mathbf{A}^{sp}$. And the corresponding weight of a hyperedge is calculated with Eqs. (2) and (14), respectively. Then, it performs as the conventional hypergraph learning process. We can obtain classification function $\mathbf{F}^*$, then the classification of i-th sample can be accomplished by assigning it to the g-th class that satisfies $g = \text{argmax} \, \mathbf{F}_{ij}^*$.

This method constructs two hyperedge sets and the corresponding weights into a new hypergraph which takes more high-order information into account. In other words, it combines two kinds of similarities: distance-similarity and linear-similarity. We consolidate the similarities between samples in the same category by putting the sparse hyperedges into the conventional hyperedges.

Strategy two

We obtain two confidences of the labeling, $\mathbf{F}$ and $\mathbf{F}^{sp}$, for conventional hypergraph learning and sparse hypergraph learning. For this strategy, we define $\mathbf{F}^*$ as

$$\mathbf{F}^* = \alpha^* \mathbf{F} + (1 - \alpha^*) \mathbf{F}^{sp}$$

$$\alpha^* = \arg\min_{\alpha} \sum_{i=1}^{m} \|\alpha \mathbf{F}_i + (1-\alpha)\mathbf{F}_i^{sp} - \mathbf{y}_i\|_2^2$$

subject to $\alpha \in [0, 1]$.   (16)

It means that, here, we combine linearly only the results of the two hypergraph learnings with the weights a^* and $1 - a^*$. We can get a^* by minimizing the loss of classification contribution value to truth classification value with only the labeled data point. The minimized function is easy to be solved since there is only one variable. After obtaining $\mathbf{F}^*$, we set the g-th class to the i-th data point if the g-th component is the maximum in the i-th row of $\mathbf{F}^*$.

For this strategy, we learn the confidence of the labeling independent for the two hypergraph-based learnings. Then, we combine only the two confidences of the labeling linearly with optimal weights.

3. Experiments

We performed experiments for image classification on two datasets: MNIST handwritten digit image data set [49] and ORL face image data set [11]. In order to assess the performance of the proposed method, we compared our method with general classification algorithms including transductive support vector machine (TSVM) [10], Simple-Graph Learning (SGL) [50], Hypergraph Learning (HL) [37], Sparse Hypergraph Learning (SHL) [53] and Semi-supervised Discriminant Analysis (SDA) [51] classifications. And more, we analyzed our method in two strategies.

3.1. Datasets and configurations

The MNIST handwritten digit image data set contains 10,000 images. There are ten classes (digit "0" to "9") of images, each class has about 1000 images, and each image is 28×28 in resolution, which results in a 784-D feature vector. For this dataset, in experiments, we randomly select 400 labeled samples as training samples and the remaining samples as testing samples. The ORL face image data set contains 64 different images for each of the 38 subjects. Each image is 48×42 in resolution, which results in a 2016-D feature vector. Images are taken at different times, with varied lighting, facial expressions, and facial details. In experiments of this dataset, we randomly select 20 labeled samples of each category as training samples and the remaining samples as testing samples. Fig. 2 shows sample images from the two data sets.

For all the classification methods, we independently repeat the experiments twenty times with randomly selected training samples and show the averaged results by Mean Average Precision (MAP).

1) Simple-graph Learning addresses the pair-wise relationships between any vertices in the correlated Laplacian graph. We use the regulation, which is similar to the first two items of Eq. (6). However, simple graph learning has difficulty tuning

the parameters of λ and μ. Here, we set λ to be the norm variance of the data points, and μ to be $\lambda/4$.

2) Semi-supervised Discriminated Analysis aims to find a projection which is in respect to the discriminated structure inferred from the training data. The training data, combined with the unlabeled data, are used to build a Laplacian graph which provides a discrete approximation to the local geometry of the data manifold and can be incorporated into objective functions. It can preserve the manifold structure. We set the weight, $w_{ij} = \exp(-\text{norm}(n_i - n_j)/2\sigma^2)$, to an edge if nodes i and j are connected. And we set neighborhood to be 5 on the k-Nearest Neighbor classification method.

3) For the method of Transductive Support Vector Machine with RBF kernel, we use the one-versus-all strategy to solve the multiclass problems. The optimal values of the radius parameter and the weighting parameter in regularization are used. As we chose C-SVC as the type of SVM, we set the parameter C to 20. We also set gamma and coefficient 0 in kernel function to 1 and set the degree to 4. And Unlike SVM, Transductive SVM has an additional parameter that modulates the effect which we set to 100.

4) Hypergraph-based learning. The neighborhood size is set to 5. And we set the iteration time in the alternating optimization process to 10. For Combinative Hypergraph Learning, we initialize a to 0.01 and iteration step by 0.01.

3.2. Experimental results

Fig. 3 shows the Mean Average Precision (MAP) with several methods on ORL and MNIST datasets. It shows that CHL achieves the best results in all the methods. And the results demonstrate the effectiveness of the proposed method. Compared with SGL, hypergraph-based learning methods achieve better results, and
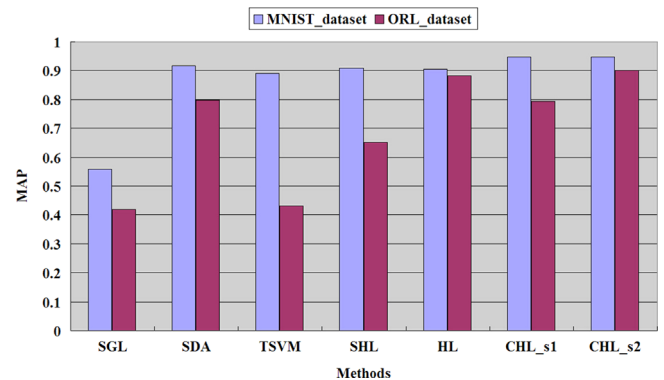
Fig. 3. Mean Average Precision (MAP) of methods for SGL, SDA, TSVM, SHL, HL, CHL_s1 and CHL_s2 on two datasets.
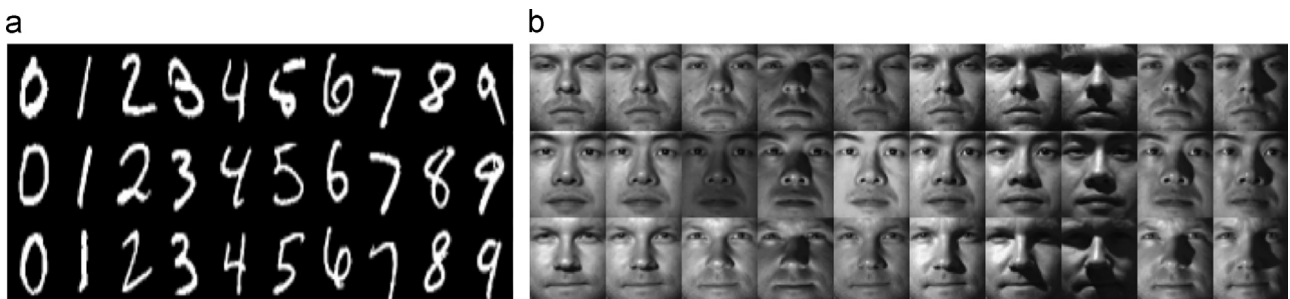
Fig. 2. Sample images from (a) MNIST and (b) ORL data sets used in the experiment.

it demonstrates that hypergraph-based learning considering high-order information is effective. SDA shows better results than TSVM and SHL. For MNIST dataset, CHL_s1 achieves better results by comparing with SDA and HL, while it shows inferior performance for ORL dataset. Both CHL_s1 and CHL_s2 show the best performance for MNIST dataset in all the methods. We affirm that CHL_s2 shows the best performance over all methods for two datasets. For CHL_s1, we concatenate the hypergraph edge sets of hypergraph learning and sparse hypergraph learning with the same weight. However, the experimental results showed that this process is not effective for ORL dataset, which demonstrates that the two hypergraph learning methods are not the same contribution for classification.

For further illustration of our proposed methods, we show the correction ratio in detail for each trial on hypergraph-based methods in Table 2. The correct ratio of HL is better than that of SHL significantly with gaining about 0.24 for ORL dataset, but, they show similar performance for MNIST dataset. We can conclude that the performance of HL is better than that of SHL for the two datasets. By combining the two hypergraph-based learnings with two strategies, the learning performance gets an overall better level than that with conventional hypergraph learning or sparse hypergraph learning independently. However, the performance of the two strategies shows different results. Both CHL_s1 and CHL_s2 achieve better results than HL and SHL for MNIST dataset, while CHL_s1 achieves worse results than that of HL for ORL dataset. For each trial on the ORL dataset, the correct ratio of CHL_s2 is a little higher than that of HL all the time, and it is large when the correct ratio of HL is low. However, the case on MNIST dataset is not the same as the ORL dataset that sometimes the correct ratio of CHL_s2 is higher than that of CHL_s1, sometimes it is inverse. Overall, they show a similar performance. It is definite that at least one strategy achieves better results than that of HL and SHL.

Furthermore, we observe the variable a^* in Eq. (15) for each trial. It is local at 0.84 for large probability for ORL dataset and at 0.9 for MNIST dataset. This demonstrates that the weight of distance-similarity is larger than that of linear-similarity, namely, that the contribution of distance-similarity on image classification is larger than that of linear-similarity in our proposed methods.

4. Conclusion

In this paper, we present a new transductive learning method called combinative hypergraph learning (CHL). CHL captures the similarity between two samples in the same category by adding sparse hypergraph learning to conventional hypergraph learning. And we present two strategies to combine the two hypergraph-based learnings. Experimental results on two image data sets have demonstrated the effectiveness of CHL in comparison to the state-of-the-art methods. However, the stability of the two combined strategies should be enhanced. In our future work, we want to analyse further the differences between CHL_s1 and CHL_s2. We perform the experiments on more datasets and show the suitable application on our proposed method. We want to provide the details of which data structure is suitable for which one method.

Acknowledgment

Table 2
Correct ration of hypergraph-based learning for two datasets.

Serial number	Correct ratio for ORL dataset				Correct ratio for MNIST dataset			
	SHL	HL	CHL_S1	CHL_S2	SHL	HL	CHL_S1	CHL_S2
1	0.660	0.956	0.940	0.960	0.909	0.908	0.946	0.945
2	0.764	0.891	0.827	0.916	0.896	0.935	0.961	0.961
3	0.643	0.931	0.824	0.935	0.927	0.865	0.941	0.940
4	0.782	0.946	0.861	0.947	0.881	0.852	0.903	0.904
5	0.637	0.644	0.530	0.764	0.891	0.886	0.931	0.932
6	0.477	0.834	0.732	0.840	0.894	0.916	0.948	0.947
7	0.678	0.947	0.834	0.954	0.884	0.912	0.917	0.917
8	0.760	0.873	0.803	0.901	0.898	0.919	0.947	0.947
9	0.776	0.894	0.831	0.912	0.899	0.895	0.951	0.950
10	0.591	0.933	0.827	0.934	0.885	0.955	0.931	0.930
11	0.416	0.785	0.757	0.798	0.933	0.874	0.973	0.922
12	0.703	0.960	0.861	0.961	0.940	0.894	0.953	0.960
13	0.421	0.697	0.559	0.713	0.890	0.927	0.936	0.947
14	0.733	0.910	0.825	0.933	0.917	0.937	0.962	0.969
15	0.577	0.922	0.756	0.931	0.915	0.920	0.964	0.981
16	0.690	0.948	0.877	0.952	0.896	0.908	0.952	0.942
17	0.672	0.940	0.847	0.942	0.896	0.894	0.944	0.970
18	0.769	0.829	0.741	0.903	0.899	0.921	0.943	0.973
19	0.519	0.845	0.790	0.852	0.945	0.954	0.969	0.982
20	0.795	0.949	0.846	0.949	0.891	0.895	0.963	0.938
Average	0.651	0.882	0.793	0.900	0.904	0.908	0.947	0.948

References

[1] M. Wu and B. Scholkopf, 2007, Transductive classification via local learning regularization, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 628–635.

[2] J. Yu, D. Liu, D. Tao, S.-H. Soon, Complex object correspondence construction in 2D animation, IEEE Trans. Image Process. 20 (11) (2011) 3257–3269.

[3] O. Chapelle, A. Zien, B. Scholkopf, Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.

[4] X. Zhu, "Semi-supervised learning literature survey," Univ. WisconsinMadison, Tech. Rep., Madison, 2005.

[5] G. Druck, A. McCallum, High-performance semi-supervised learning using discriminatively constrained generative models, in: Proceedings of the International Conference on Machine Learning, Haifa, Israel, 2010, pp. 319–326.

[6] R. He, W. Zheng, Nonnegative sparse coding for discriminative semi-supervised learning, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, 2011, pp. 2849–2856.

[7] P. Mallapragada, R. Jin, A. Jain, Y. Liu, SemiBoost: Boosting for semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2000–2014.

[8] C. Rosenberg, M. Heberg, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of the Workshop on Applied Computer Vision, Breckenridge, CO, 2005, pp. 29–36.

[9] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Workshop on Computational Learning Theory, Madison, WI, 1998, pp. 92–100.

[10] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the International Conference on Machine Learning, Bled, Slovenia, 1999, pp. 200-209.

[11] D. Zhou, O. Bousquet, T. Lal, J. Weston, B. Scholkopf, Learning with local and global consistency, in: Proceedings of the Neural Information Processing System, Vancouver, BC, Canada, 2004, pp. 321–328.

[12] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.

[13] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 55–67.

[14] S. Agarwal, J. Lim, L. Zelnik Manor, P. Perona, D. Kriegman, S. Belongie, Beyond pairwise clustering, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005, pp. 838–845.

[15] R. Zass, A. Shashua, Probabilistic graph and hypergraph matching, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition in Anchorage, AK, 2008, pp. 1–8.

[16] L. Sun, S. Ji, J. Ye, Hypergraph spectral learning for multi-label classification, in Proceedings of the International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, 2008, pp. 668–676.

[17] Y. Huang, Q. Liu, D. Metaxas, Video object segmentation by hypergraph cut, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 1738–1745.

[18] Z. Tian, T. Hwang, R. Kuang, A hypergraph-based learning algorithm for classifying gene expression and array CGH data with prior knowledge, Bioinformatics 25 (21) (2009) 2831–2838.

[19] Y. Huang, Q. Liu, S. Zhang, and D. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 3376–3383.

[20] A. Wong, S. Lu, Recognition and shape synthesis of 3D objects based on attributed hypergraphs, IEEE Trans. Pattern Anal. Mach. Intell. 11 (3) (1989) 279–290.

[21] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, X. He, Music recommendation by unified hypergraph: combining social media information and music content, in: Proceedings of the ACM International Conference on Multimedia, Firenze, Italy, 2010, pp. 391–400.

[22] N. Guan, D. Tao, Z. Luo, B. Yuan, Manifold regularized discriminative non-negative matrix factorization with fast gradient descent,, IEEE Trans. Image Process. 20 (7) (2011) 2030–2048.

[23] X. Zhu, Z. Ghaharmani, J. Lafferty, Semisupervised learning using Gaussian fields and harmonic functions, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 912–919.

[24] M. Belkin, L. Matveeva, P. Niyogi, Regularization and semisupervised learning on large graphs, in: Proceedings of the Conference on the Computer Learning Theory, Banff, Canada, 2004, pp. 624–638.

[25] H. Shin, J. Hill, G. Ratsch, Graph-based semi-supervised learning with sharper edges, in: Proceedings of the European Conference on Machine Learning, Berlin, Germany, 2006, pp. 401–412.

[26] X. Zhu, Semi-supervised learning with graphs, (M.S. thesis), Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2005, CMULTI-05-192.

[27] M. Culp, G. Michailidis, Graph based semi-supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 30 (1) (2008) 174–179.

[28] W. Liu, J. He, S. Chang, Large graph construction for scalable semi-supervised learning, in: Proceedings of the International Conference on Machine Learning, Haifa, Israel, 2010, pp. 679–686.

[29] J. Yu, M. Wang, D. Tao, Semi-supervised multiview distance metric learning for cartoon synthesis, IEEE Trans. Image Process. 21 (11) (2012) 4636–4648.

[30] J. Yu, D. Tao, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. PP (99) (2014) 1. http://dx.doi.org/10.1109/TCYB.2014.2336697.

[31] J. Yu, D. Tao, J. Li, J. Cheng, Semantic preserving distance metric learning and applications, Inf. Sci. 281 (2014) 674–686. http://dx.doi.org/10.1016/j.ins.2014.01.025.

[32] M. Wang, X. Hua, J. Tang, R. Hong., Beyond distance measurement: constructing neighborhood similarity for video annotation, IEEE Trans. Multimed. 11 (3) (2009) 465–476. http://dx.doi.org/10.1109/TMM.2009.2012919.

[33] M. Wang, X. Hua, R. Hong, J. Tang, G.-J. Qi, Y. Song, Unified video annotation via multi-graph learning, IEEE Trans. Circuits Syst. Video Technol. 19 (5) (2009) 733–746. http://dx.doi.org/10.1109/TCSVT.2009.2017400.

[34] M. Wang, H. Li, D. Tao, K. Lu, X. Wu, Multimodal graph-based reranking for web image search, IEEE Trans. Image Process. 21 (11) (2012) 4649–4661. http://dx.doi.org/10.1109/TIP.2012.2207397.

[35] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3D object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303. http://dx.doi.org/10.1109/TIP.2012.2199502.

[36] Y. Gao, M. Wang, Z. Zha, J. Shen, X. Li, X. Wu, Visual-textual joint relevance learning for tag-based social image search, IEEE Trans. Image Process. 22 (1) (2013) 363–376. http://dx.doi.org/10.1109/TIP.2012.2202676.

[37] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: clustering, classification, and embedding, in: Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 2006, pp. 1601–1608.

[38] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 21 (2012) 3262–3272.

[39] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032. http://dx.doi.org/10.1109/TIP. 2311377.

[40] J. Yu, Y. Rui, Y. Tang, D. Tao, High-order distance based multiview stochastic learning in image classification, IEEE Trans. Cybern. 44 (12) (2014) 2431–2442. http://dx.doi.org/10.1109/TCYB.2014.2307862.

[41] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multiview features for image reranking, IEEE Trans. Multimed. 16 (2014) 159–168.

[42] J. Marial, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, CVPR 1-12 (2008) 2415–2422.

[43] J. Marial, M. Elad, G. Sapiro, Sparse representation for color image restoration, TIP 17 (1) (2008) 53–69.

[44] S. Rao, R. Tron, Y. Ma, R. Vidal. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[45] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE. Trans. Pattern. Anal. Mach. Intell. 31 (2) (2009).

[46] Emmanuel J. Candes, Terence Tao, Decoding by linear programming, IEEE Trans. Inf. Theory 51 (12) (2005) 4203–4215.

[47] J. Candès Emmanuel, The restricted isometry property and its implications for compressed sensing, Theory Signals/Math. Anal. Ser. I 346 (2008) 589–592.

[48] Binghui Wei, Jun Yu, Cheng Wang, Jonathan Li, PolSAR image classification using a semi-supervised classifier based on hypergraph learning, Remote Sens. Lett. 5 (2014) 386–395.

[49] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification," in: Proceedings of Workshop Applied Computer Vision, Sarasota, FL, 1994, pp. 138–142.

[50] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: Proceedings of the EEE 11th International Conference on Computer Vision, 2007, pp. 1–7.

[51] Y. Huang, Q. Liu, F. Lv, Y. Gong, D. Metaxas, Unsupervised image categorization by hypergraph partition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (6) (2011) 1266–1273.

[52] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, Laplacian sparse coding, hypergraph laplacian sparse coding, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 92–104.

[53] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of the IEEE-computer-society conference on computer vision and pattern recognition, vol. 1–4, Miami Beach, FL, pp. 2782–2789, June 20–25, 2009.

[54] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE. Trans. Pattern. Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
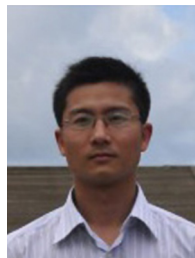
Binghui Wei received the M.S. degree in computer science from Jiangxi University of Science and Technology, Ganzhou, China, in 2010. He is currently working toward the Ph.D. degree in the Department of Computer Science, Xiamen University. His research interests include computer vision, machine learning, remote sensing image processing.

Ming Cheng received the B.E. degree from the Department of Electrical Engineering, Tsinghua University, China, in 1999, and the Ph.D. degree from the Department of Biomedical Engineering, Tsinghua University, China, in 2004. He is an associate professor in the Department of Computer Science, Xiamen University, China. His main research areas are computer vision, remote sensing image processing, and LiDAR point cloud processing.

Cheng Wang received the Ph.D. degree in information and communication engineering from National University of Defense Technology, Changsha, China, in 2002. He is a Professor and the Associate Dean of the School of Information Science and Technology, Xiamen University, Xiamen, China. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion. He is a Cochair of the working group I/3 of the International Society for Photogrammetry and Remote Sensing.

Jonathan Li received the Ph.D. degree in geomatics engineering from University of Cape Town, Cape Town, South Africa. He is with the Ministry of Education Key Laboratory of Underwater Acoustic Communication and Marine Information Technology, Xiamen University, Xiamen, China. He has co-authored over 200 publications, over 60 of which were published in refereed journals. His research interests include information extraction from Earth observation images and 3-D surface reconstruction from mobile LiDAR point clouds.