SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2018, Vol. 61 112209:1–112209:14
https://doi.org/10.1007/s11432-017-9421-3

Discriminative graph regularized broad learning system for image recognition

Junwei JIN[1], Zhulin LIU[1] & C. L. Philip CHEN[1,2,3*]

[1]*Faculty of Science and Technology, University of Macau, Macau* 999078*, China;*
[2]*Dalian Maritime University, Dalian* 116026*, China;*
[3]*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100080*, China*

Received 12 December 2017/Revised 8 February 2018/Accepted 30 March 2018/Published online 17 October 2018

Abstract Broad learning system (BLS) has been proposed as an alternative method of deep learning. The architecture of BLS is that the input is randomly mapped into series of feature spaces which form the feature nodes, and the output of the feature nodes are expanded broadly to form the enhancement nodes, and then the output weights of the network can be determined analytically. The most advantage of BLS is that it can be learned incrementally without a retraining process when there comes new input data or neural nodes. It has been proven that BLS can overcome the inadequacies caused by training a large number of parameters in gradient-based deep learning algorithms. In this paper, a novel variant graph regularized broad learning system (GBLS) is proposed. Taking account of the locally invariant property of data, which means the similar images may share similar properties, the manifold learning is incorporated into the objective function of the standard BLS. In GBLS, the output weights are constrained to learn more discriminative information, and the classification ability can be further enhanced. Several experiments are carried out to verify that our proposed GBLS model can outperform the standard BLS. What is more, the GBLS also performs better compared with other state-of-the-art image recognition methods in several image databases.

Keywords broad learning system, deep learning, graph regularization, image recognition, feature extraction, incremental learning

Citation Jin J W, Liu Z L, Chen C L P. Discriminative graph regularized broad learning system for image recognition. Sci China Inf Sci, 2018, 61(11): 112209, https://doi.org/10.1007/s11432-017-9421-3

1 Introduction

Image recognition, aiming at predicting the label of one (or multiple) query image, has been a fundamental problem in pattern classification and computer vision [1, 2]. Various applications on recognition are growing rapidly and extensively in our daily life. Generally speaking, a representative image recognition system consists of two essential components, namely feature extraction and pattern classification [3]. The past few years have witnessed a rapid progress of the image recognition, mainly due to the success of deep learning technology [4, 5]. Nevertheless, in deep learning architecture, there exists too many hyperparameters which make the training process time-consuming and the complicated structure leads to difficulties in theoretical analysis. These drawbacks seriously discourage the further applications of deep learning in various fields, besides image recognition.

* Corresponding author (email: philip.chen@ieee.org)

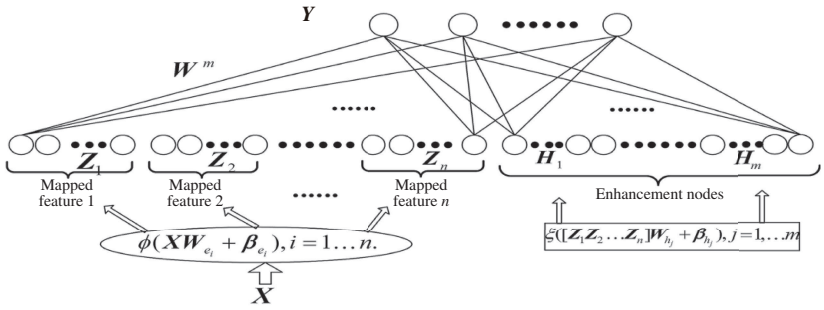

Figure 1 The structure of BLS. First, the input features are randomly mapped into a series of features spaces. Second, the random features are transformed to the enhancement nodes. At the output layer, all the features are connected together to link the label layer.

The random vector functional link neural networks (RVFLNN) [6–8] provides another representative method to learn the parameters in neural network. It is based on the architecture of functional link neural network (FLNN) [9]. However, like traditional single layer neural network, FLNN depends generally on the back propagation algorithm to iteratively train all the parameters. With the number of hidden layers increases, this kind of parameters learning method will cause many problems such as local minima, time-consuming and slow convergence. To overcome these drawbacks, FLNN with random weights (RVFLNN) was proposed in which the input weights are randomly generated in a suitable domain and stored to do the testing. In [10], the author proved the universal approximation of RVFLNN on the compact set. Then various improvement studies of RVFLNN are developed [8, 11]. Combined with the supervised learning algorithm, Chen et al. [12, 13] proposed the rank-expansion with instant learning to implement the RVFLNN adaptively. In [14], a dynamic step-wise updating algorithm was proposed to make the RVFLNN flexible for the modern large data era [15].

Broad learning system (BLS) is an emerging technology proposed by Chen et al. [16]. Its complete schematic is shown in Figure 1. It is obvious that the BLS is established on the idea of RVFLNN. Specifically, the main characteristics of BLS are as follows: firstly, the input data is transformed into numbers of feature nodes by proper mapping functions, and then these features are concatenated together to generate a series of enhancement nodes randomly, which will be used to expand the network architecture in a broad sense. Last but not least, all the features nodes and the outputs of the enhancement nodes are connected together to feed into the output layer. The desired output weights are to be determined by a fast ridge regression of the pseudo-inverse of the system equation. We can see that except output weights, the other weights and biases in BLS are all randomly generated. In addition, the incremental learning algorithm is incorporated into the BLS which makes the network can be remodeled fast in the broad expansion without a retraining process if the network deems to be expanded. Therefore, BLS is an ideal architecture for modeling and learning in time-variant big data environment. In [17], the universal approximation property and various structural variations of BLS are further presented. Combining with fuzzy logic, Feng and Chen [18] proposed a novel fuzzy broad learning system for regression and classification. Through these researches in theory and practical applications, it can be concluded that BLS indeed significantly outperforms the existing deep learning method and other network structure in learning accuracy and generalization ability. More details of BLS we refer to [16].

In the past decades, learning with manifold structure of data has attracted much attention of researchers, such as LLE, MDA, Laplacian Eigenmap, ISOMAP [19, 20], One of the key points they share is to preserve the locally invariant of data, i.e., they are likely to have the similar conditional probabilities if the points are close to each other. From this perspective, we can see that the objective of BLS in linear regression form mainly focuses on approximating the desired training labels, and the underlying geometrical structure of data is not fully considered. The output weights learnt in this way cannot vary smoothly along the geodesics of the data manifold. Worse, the performance of classification will be affected. Recently, various researchers have considered the case when data is drawn from sampling a probability distribution that has support on or near to a submanifold of the ambient space. And the local

consistency property has been applied to be a side information for improving the performance of learning models. Following this idea, in this paper, we propose a novel extension of BLS model which incorporates the criteria employed of the manifold leaning. The optimization process of proposed model will be formulated to minimize both the training error of network and manifold learning criterions simultaneously. This can be done by incorporating an appropriate graph regularization term on the objective of the standard BLS, named as graph regularized broad learning system (GBLS). The graph regularization term is constructed within the graph embedding framework [21], in which the training data is assumed to form the vertex set of an undirected graph G with a weight matrix to express the similarity between the vertices. Moreover, a penalty graph G^p can also be defined, whose corresponding weight matrix penalizes specific characteristics of the relationships between the training data. By using such an approach, the learnt output weights in GBLS can be more discriminative. Moreover, there exists an analytical solution for the objective function of GBLS, so the optimization process is as efficient as BLS.

The remainder of this paper is outlined as follows. We introduce the standard BLS in Section 2. Section 3 presents our proposed GBLS model and the corresponding algorithm. In Section 4, experiments are conducted to validate the proposed GBLS on popular face databases and other visual databases. Finally, Section 5 concludes this paper.

2 Broad learning system

BLS is a novel network architecture, proposed by Chen et al. [16], to provide an effective and efficient learning framework for machine learning and pattern recognition. In contrast to the deep learning algorithms which consume too much time to train abundant parameters in the filters and layers, the parameters in BLS could be determined either by random projection or a fast pseudo-inverse technology. What is more, the training process can be extended to an incremental learning mode in which the retraining process is no longer necessary when the network structure is changed by new nodes or input.

Now let us express the BLS mathematically. Considering the general supervised learning task, we are given the training data set $\{(\boldsymbol{X}, \hat{\boldsymbol{Y}}) | \boldsymbol{X} \in \mathbb{R}^{N\times D}, \boldsymbol{Y} \in \mathbb{R}^{N\times C}\}$ from C classes, where each row in $\boldsymbol{X}$ and $\boldsymbol{Y}$ denotes the data point $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and target vector $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iC})$, respectively. In BLS, the training samples are first transformed into N_w random feature spaces by N_w feature mapping function $\phi_i, i = 1, 2, \ldots, N_w$, that is,

$$\boldsymbol{Z}_i = \phi_i(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i}), \quad i = 1, 2, \ldots, N_w, \tag{1}$$

where the weights $\boldsymbol{W}_{e_i}$ and the bias term $\boldsymbol{\beta}_{e_i}$ are randomly determined with the proper dimensions.

Then, we define the feature space of training samples as $\boldsymbol{Z}^{N_w} \triangleq [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_{N_w}]$, a collection of N_w groups of feature nodes. Just like the FLNN, $\boldsymbol{Z}^{N_w}$ should be applied to generate and connect the layer of enhancement nodes. To speed up the training process, the enhancement nodes are obtained group by group. The outputs of the j-th group of enhancement nodes are defined by

$$\boldsymbol{H}_j \triangleq \xi_j(\boldsymbol{Z}^{N_w}\boldsymbol{W}_{h_j} + \boldsymbol{\beta}_{h_j}), \ j = 1, 2, \ldots, m, \tag{2}$$

where ξ_j is a nonlinear activation function. And we denote the outputs of the enhancement layer by $\boldsymbol{H}^m \triangleq [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]$.

For simplicity and without loss of generality, we will omit the subscripts of the feature mapping ϕ_i and the activation function ξ_j in the following part. However, the ϕ_i can be selected differently, as well as the ξ_j. In order to obtain sparse representation of input data, the randomly initialized weight matrix $\boldsymbol{W}_{e_i}$ can be tuned by applying the linear inverse problem (please refer to (4) in [16]).

Therefore, the output $\hat{\boldsymbol{Y}}$ of a BLS has the following form:

$$\hat{\boldsymbol{Y}} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n, \boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]\boldsymbol{W} = [\boldsymbol{Z}^n, \boldsymbol{H}^m]\boldsymbol{W} = \boldsymbol{A}\boldsymbol{W}, \tag{3}$$

where $\boldsymbol{A} = [\boldsymbol{Z}^n, \boldsymbol{H}^m]$, and $\boldsymbol{W}$ is the output weight connecting the feature nodes and enhancement nodes to the output layer. $\boldsymbol{W}$ could be obtained by minimize the objective

$$\arg\min_{\boldsymbol{W}} \ \mathcal{J}_{\mathrm{BLS}} = \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W}\|^2 + \lambda\|\boldsymbol{W}\|^2, \tag{4}$$

where the first term denotes the training errors, the second term is a regularization term which controls the complexity of the model. And λ is a regularization parameter to balance the influence of error term and the model complexity. Through a simple derivative operation on $\boldsymbol{W}$, we can obtain

$$\boldsymbol{W} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}. \tag{5}$$

The calculation of BLS output weights $\boldsymbol{W}$ can always be achieved, as the matrix $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda\boldsymbol{I})$ is generally nonsingular.

3 The proposed graph regularized BLS

In this section, we focus on our proposed GBLS model and the corresponding algorithm. After randomly generation of the input and enhancement weights, the output weights of BLS can be solved analytically by (5). The training error can be minimized as small as possible, however, the underlying geometrical structure of data is ignored. To overcome this drawback, we propose the GBLS which incorporates the manifold technology into the optimization process of BLS. Following the general manifold regularization method, the objective function of GBLS should have the following form:

$$\arg\min_{\boldsymbol{W}} \ \mathcal{J}_{\mathrm{GBLS}} = \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W}\|^2 + \lambda_1 E_G + \lambda_2\|\boldsymbol{W}\|^2, \tag{6}$$

where E_G is the graph regularization term to reflect the local discriminative structure of data. λ_1 and λ_2 are two trade off parameters dictating the importance of $\|\boldsymbol{W}\|^2$ and E_G, respectively. Then the key point is to construct an appreciate graph regularization term E_G to make the output weights to be more discriminative.

The E_G is constructed within the graph embedding framework in [21]. In our GBLS model, the training data $\boldsymbol{X}$ is viewed as the vertex set of an undirected weight graph $G(\boldsymbol{X}, \boldsymbol{V})$, where $\boldsymbol{V} \in \mathbb{R}^{N\times N}$ is an adjacent matrix recording the similarity between $\boldsymbol{X}$. Further, a penalty graph $G^p(\boldsymbol{X}, \boldsymbol{V}^p)$ can be defined, whose weight matrix $\boldsymbol{V}^p \in \mathbb{R}^{N\times N}$ penalizes specific relationships between the graph vertices x_i. The geometric structure of training data can be reflected well by proper graphs $G(\boldsymbol{X}, \boldsymbol{V})$ and $G^p(\boldsymbol{X}, \boldsymbol{V})$. There exists various successful manifold learning algorithms, such as LLE, MDA, and Laplacian Eigenmap. In fact, the above algorithms are all inspired by the same assumption of manifold learning: the nearby points are likely to have similar embedding. Nevertheless, different graph construction ways that appear in these methods lead to different performances and application ranges. Based on two different prior knowledge, we are able to consider two different graph structures of data leading to two GBLS models: IGBLS which combines the intrinsic graph with BLS, and IPGBLS which considers the intrinsic and penalty graph simultaneously into BLS.

3.1 IGBLS: BLS exploiting intrinsic graph

In IGBLS model, we consider the intrinsic graph of data, and the E_G is used to express intrinsic relationships of training data. All the training data are used to form the vertex set of a graph $G(\boldsymbol{X}, \boldsymbol{V})$, where $\boldsymbol{V}$ is the similarity matrix whose elements denote the relationships between the graph vertices x_i. We assume the data from the same class is from the same manifold space, while different classes are located in different manifolds. That means the manifold structure of data in this case we consider is on class level. Generally, there are two kinds of methods: KNN and ϵ-ball, and the weights can be calculated by the Gaussian heat kernel distance or binary weight [19, 22]. However, both of them determine the neighboring samples based on pairwise Euclidean distance, which is very sensitive to data noise and one noisy feature may dramatically change the graph structure. Moreover, the distance information among

samples are destroyed by nonlinear mapping in BLS. Therefore, here the supervised version of graph construction should be applied. The element of $\boldsymbol{V}$ is defined on the label information as follows:

$$V_{ij} = \begin{cases} 1/N_t, & \text{if } l(x_i) = l(x_j), \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where N_t denotes the sample number of the t-th class, $l(x_i)$ is the label of sample x_i.

By using (7), the graph regularization term E_{IGBLS} can be formulated as

$$\sum_{ij} V_{ij} \|\hat{y}_i - \hat{y}_j\|^2 = \text{Tr}(\hat{\boldsymbol{Y}}^{\text{T}} \boldsymbol{L}_{\text{IGBLS}} \hat{\boldsymbol{Y}}), \tag{8}$$

where $\hat{y}_i$ and $\hat{y}_j$ are the predictions with respect to samples x_i and x_j, respectively. $\text{Tr}(\cdot)$ is the trace operator of a matrix, $\boldsymbol{L}_{\text{IGBLS}} \in \mathbb{R}^{N \times N}$ is the so-called graph Laplacian matrix defined as $\boldsymbol{L}_{\text{IGBLS}} = \boldsymbol{D} - \boldsymbol{V}$. $\boldsymbol{D}$ denotes a diagonal degree matrix which can be obtained by $\boldsymbol{D}_{ii} = \sum_j V_{ij}$. To reach better performance, a normalization form such as $\boldsymbol{D}^{-1/2} \boldsymbol{L}_{\text{IGBLS}} \boldsymbol{D}^{-1/2}$ or an alternative modification $\boldsymbol{L}^k_{\text{IGBLS}}$ (k is an integer to be set) would be adopted instead of using the $\boldsymbol{L}_{\text{IGBLS}}$ directly.

In this way, the IGBLS model can be optimized by the objective as

$$\arg\min_{\boldsymbol{W}} \ \mathcal{J}_{\text{IGBLS}} = \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W}\|^2 + \lambda_1 \text{Tr}(\hat{\boldsymbol{Y}}^{\text{T}} \boldsymbol{L}_{\text{IGBLS}} \hat{\boldsymbol{Y}}) + \lambda_2 \|\boldsymbol{W}\|^2, \tag{9}$$

where $\hat{\boldsymbol{Y}} = \boldsymbol{A}\boldsymbol{W}$ denotes the predicted labels by our algorithm. Now by substituting $\hat{\boldsymbol{Y}} = \boldsymbol{A}\boldsymbol{W}$ into $\mathcal{J}_{\text{IGBLS}}$, we can calculate $\boldsymbol{W}$ as follows:

$$\begin{aligned} \frac{\partial \mathcal{J}_{\text{IGBLS}}}{\partial \boldsymbol{W}} =& \frac{\partial}{\partial \boldsymbol{W}} \left[\text{Tr}[(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W})^{\text{T}} (\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W})] + \lambda_1 \text{Tr}(\boldsymbol{W}^{\text{T}} \boldsymbol{A}^{\text{T}} \boldsymbol{L}_{\text{IGBLS}} \boldsymbol{A}\boldsymbol{W}) + \lambda_2 \|\boldsymbol{W}\|^2 \right] \\ =& 2\boldsymbol{A}^{\text{T}}\boldsymbol{A}\boldsymbol{W} - 2\boldsymbol{A}^{\text{T}}\boldsymbol{Y} + 2\lambda_1 \boldsymbol{A}^{\text{T}} \boldsymbol{L}_{\text{IGBLS}} \boldsymbol{A}\boldsymbol{W} + 2\lambda_2 \boldsymbol{W} \\ =& 0. \end{aligned} \tag{10}$$

As a result, we can get

$$\boldsymbol{W} = (\boldsymbol{A}^{\text{T}}\boldsymbol{A} + \lambda_1 \boldsymbol{A}^{\text{T}} \boldsymbol{L}_{\text{IGBLS}} \boldsymbol{A} + \lambda_2 \boldsymbol{I})^{-1} \boldsymbol{A}^{\text{T}} \boldsymbol{Y}. \tag{11}$$

3.2 IPGBLS: BLS exploiting simultaneously intrinsic and penalty graph

In this case, E_G is to express both intrinsic and penalty training data relationships [23]. In IPGBLS model, two graphs, intrinsic graph $G^w(\boldsymbol{X}, \boldsymbol{V}^w)$ and penalty graph $G^p(\boldsymbol{X}, \boldsymbol{V}^p)$, are constructed in the original data space as follows:

$$V^w_{ij} = \begin{cases} 1, & \text{if } l(x_i) = l(x_j), \text{ and } x_i \in \mathcal{N}_{k1}(x_j), \\ 1, & \text{if } l(x_i) = l(x_j), \text{ and } x_j \in \mathcal{N}_{k1}(x_i), \\ 0, & \text{otherwise}, \end{cases} \quad V^p_{ij} = \begin{cases} 1, & \text{if } x_i \neq x_j, \text{ and } x_i \in \mathcal{N}_{k2}(x_j), \\ 1, & \text{if } x_i \neq x_j, \text{ and } x_j \in \mathcal{N}_{k2}(x_i), \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\mathcal{N}_k(x_i)$ denotes the set of k nearest neighbors of x_i, and $(k1, k2)$ denotes the number of nearest neighbors. Then the Laplacian matrix of intrinsic graph is $\boldsymbol{L}^w = \boldsymbol{D}^w - \boldsymbol{V}^w$ where $\boldsymbol{D}^w = \sum_j V^w_{ij}$, and $\boldsymbol{L}^p = \boldsymbol{D}^p - \boldsymbol{V}^p$ where $\boldsymbol{D}^p = \sum_j V^p_{ij}$. The geometric structure of data in intrinsic graph can be expressed as

$$\sum_{ij} V^w_{ij} \|\hat{y}_i - \hat{y}_j\|^2 = \text{Tr}(\hat{\boldsymbol{Y}}^{\text{T}} \boldsymbol{L}^w \hat{\boldsymbol{Y}}). \tag{13}$$

Meanwhile, to encode the discriminative information in the penalty graph, we maximize margins between different classes. According to the construction of penalty graph in (12), we can get the graph-preserving criterion of discriminative information as

$$\sum_{ij} V^p_{ij} \|\hat{y}_i - \hat{y}_j\|^2 = \text{Tr}(\hat{\boldsymbol{Y}}^{\text{T}} \boldsymbol{L}^p \hat{\boldsymbol{Y}}). \tag{14}$$

We minimize (13) to retain the data geometric structure, while we maximize (14) to make the samples in different classes separable. Thus, by combining this two objectives, we can get the following objective:

$$\arg\min \mathrm{Tr}\left(\hat{\boldsymbol{Y}}^{\mathrm{T}}((\boldsymbol{L}^p)^{-1/2})^{\mathrm{T}}\boldsymbol{L}^w((\boldsymbol{L}^p)^{-1/2})\hat{\boldsymbol{Y}}\right). \tag{15}$$

Accordingly, the E_G in this model has the following expression:

$$E_{\mathrm{IPGBLS}} = \mathrm{Tr}\left(\hat{\boldsymbol{Y}}^{\mathrm{T}}((\boldsymbol{L}^p)^{-1/2})^{\mathrm{T}}\boldsymbol{L}^w((\boldsymbol{L}^p)^{-1/2})\hat{\boldsymbol{Y}}\right). \tag{16}$$

We add a tiny perturbation to the diagonal of the graph Laplacian matrix $\boldsymbol{L}^p$, i.e., $\tilde{\boldsymbol{L}}^p = \boldsymbol{L}^p + \zeta\boldsymbol{I}$, to make it invertible. Related research has verified that the solution in this way will be consistent with the one obtained by original graph matrix, as long as the ζ is set to be small. In our experiments, ζ is empirically set to be $10^{-3}\mathrm{Tr}(\boldsymbol{L}^p)$. And we will still use the $\boldsymbol{L}^p$ to imply the perturbed matrix $\tilde{\boldsymbol{L}}^p$ for simplicity in the rest of this paper.

In this way, the IPGBLS model can be optimized by the objective as

$$\arg\min_{\boldsymbol{W}} \mathcal{J}_{\mathrm{IPGBLS}} = \|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{W}\|^2 + \lambda_1\mathrm{Tr}\left((\hat{\boldsymbol{Y}})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{IPGBLS}}(\hat{\boldsymbol{Y}})\right) + \lambda_2\|\boldsymbol{W}\|^2, \tag{17}$$

where $\boldsymbol{L}_{\mathrm{IPGBLS}} \triangleq ((\boldsymbol{L}^p)^{-1/2})^{\mathrm{T}}\boldsymbol{L}^w((\boldsymbol{L}^p)^{-1/2})$ denotes the unified graph Laplacian matrix for graphs G_w and G_b. Obviously, the output weights of IPGBLS can be calculated as

$$\boldsymbol{W} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda_1\boldsymbol{A}^{\mathrm{T}}\boldsymbol{L}_{\mathrm{IPGBLS}}\boldsymbol{A} + \lambda_2\boldsymbol{I})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Y}. \tag{18}$$

Based on the above discussion, we summarized the main steps of our proposed GBLS models in Algorithm 1.

Algorithm 1 Discriminative GBLS models

Input: training set $\{\boldsymbol{X}, \boldsymbol{Y}\}$, the feature mapping function $\phi(\cdot)$, the activation function $\xi(\cdot)$, the number of feature mapping groups N_w, feature nodes N_f, enhancement nodes N_e, regularization parameter (λ_1, λ_2) and NNs $(k1, k2)$;

Output: Output weight $\boldsymbol{W}$;

Step 1:

1. Random $\boldsymbol{W}_{e_i}$, $\boldsymbol{\beta}_{e_i}$, $i = 1, 2, \ldots, N_w$;
2. Calculate $\boldsymbol{Z}_i = \phi(\boldsymbol{X}\boldsymbol{W}_{e_i} + \boldsymbol{\beta}_{e_i})$, $i = 1, 2, \ldots, N_w$;
3. Set the feature mapping group $\boldsymbol{Z}^n = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n]$;

Step 2:

1. Random $\boldsymbol{W}_h$, $\boldsymbol{\beta}_h$;
2. Calculate $\boldsymbol{H}^m = \xi(\boldsymbol{Z}^n\boldsymbol{W}_h + \boldsymbol{\beta}_h)$;

Step 3:

1. Set $\boldsymbol{A} = [\boldsymbol{Z}^n, \boldsymbol{H}^m]$;
2. Construct the adjacent matrix $\boldsymbol{V}$ by (7) and (12); then calculate the graph regulation term by (8) and (16);
3. Calculate the output weight matrix $\boldsymbol{W}$ by (11) and (18).

3.3 Relationship between DRLSC and GBLS models as well as GELM

It is clear that our GBLS models are the natural extension of BLS with the manifold regularization, and the manifold learning methods have also been combined with other machine learning algorithms, such as DRLSC [24] and GELM [25]. As we have shown, the least square only is used as the objective function in BLS to learn the output weights discriminatively. And there exists another important blocks in BLS, such as random projection and functional link. All these blocks work collaboratively to make the BLS performance well. And in DRLSC, the inputs are thrown directly into the objective function without nonlinear feature transformation, which is different from BLS. On the other hand, the GELM is built on the ELM, which is also a popular randomized neural network. However, the architecture of ELM is completely different from the BLS's. In ELM, the inputs are simply projected to the hidden layer randomly, and then the weights are determined by the hidden nodes and the desired outputs using a least square method. Different architecture in neural network leads to different recognition performance, we will show the comparison results in Section 4.

(a) (b) (c)

Figure 2 Sample from three face databases. Samples of three subjects in (a) the ORL databases, (b) the ExYaB database, and (c) the UMIST database.

4 Experiments

In this section, we comprehensively evaluate the performance of our proposed approaches from different aspects. In Subsection 4.1, we compare the two GBLS models with the standard BLS on three popular face databases: ORL, Extended Yale B, and UMIST database. In Subsection 4.2, two face databases: ExYaB, LFW [26], and four other visual recognition databases: Standford 40, CUB200-2011, Oxford 102, and Caltech-256 [3], are used to compare the two GBLS models with many other state-of-the-art classifiers. We apply the source code of the competing classifiers from their authors, and all the parameters in these classifiers are tuned to achieve the best recognition performance in all experiments. All the experiments are carried out using MATLAB on a 3.40 GHz machine with 15.90 GB RAM.

4.1 Comparison with the standard BLS

In this part, we focus on comparing our proposed IGBLS and IPGBLS model with the standard BLS. The detailed description of three databases: ORL , ExYaB and UMIST used in this part are as follows.

• ORL databse. The database contains 10 different face images of 40 subjects. For each subject, the images are taken against a dark homogeneous background with the subject in different positions. The face images are taken at different times, varying illumination, expressions and facial details. The grey-scale facial images are cropped with the size 32×32.

• ExYaB database. This database consists of 2414 frontal-face images of 38 subjects and each subject has around 64 near frontal images under different illuminations and expressions. The original images were cropped to have 32×32 pixels.

• UMIST Database. This database is composed of 575 images of 20 distinct subjects with resolution 112×92. This database is more challenging because of the larger variations between the face images of the same face in viewing direction than regular image variations in subject identity. We resize each image to 56×46 pixels.

For each database, three subjects with five sample images are randomly chosen and illustrated in Figure 2.

Each database is divided into various training and testing sets. We randomly select $\mathrm{Tn} = \{5, 6, 7\}$ (ORL), $\mathrm{Tn} = \{10, 20, 30\}$ (ExYaB) and $\mathrm{Tn} = \{5, 10, 15\}$ (UMIST) images from each subject for training and the rest are used for testing. To get a fair and better estimation of recognition accuracy, this division process is repeated 10 times. We compute and report the mean recognition accuracy value at last. The original vectored grey images, without doing dimensionality reduction, are directly applied to perform recognition. From the algorithms of different BLS models, we can see that there are three same parameters: the number of mapping groups N_w, feature nodes N_f and enhancement nodes N_e. The appreciate grid search for these three parameters is performed on different experiments to make the model achieve the best performance. The sensitivity of the other parameters will be discussed in Subsection 4.3.

Tables 1 and 2 show the parameters setting, recognition results and the running time of different BLS models on different databases, respectively. From Table 2, we can see that all different BLS models can attain 100% training accuracy on all the databases with different training sets. Hence the difference of testing accuracy among them should be crucial to compare the performance. Obviously, the incorporation of graph structure into the BLS optimization process indeed enhance the recognition performance, both

Table 1 Parameters setting of BLS, IGBLS, IPGBLS on different databases

	Number	BLS			IGBLS			IPGBLS		
		N_w	N_f	N_e	N_w	N_f	N_e	N_w	N_f	N_e
ORL	Tn = 5	20	25	500	15	20	380	20	20	400
	Tn = 6	10	26	460	12	16	400	20	20	640
	Tn = 7	20	20	400	20	20	240	22	20	500
ExYaB	Tn = 10	30	60	3000	20	50	2000	40	50	1300
	Tn = 20	30	60	4000	20	50	3000	30	35	3000
	Tn = 30	30	60	5000	30	40	2000	34	40	2000
UMIST	Tn = 5	10	9	400	11	9	900	11	9	900
	Tn = 10	30	20	300	15	10	300	12	9	860
	Tn = 15	10	9	575	10	9	300	11	11	400

Table 2 Recogniton results and running time of standard BLS, IGBLS, IPGBLS on different databases[a)]

	Number	BLS			IGBLS			IPGBLS		
		Training (%)	Testing (%)	Time (s)	Training (%)	Testing (%)	Time (s)	Training (%)	Testing (%)	Time (s)
ORL	Tn = 5	100	95.00	0.44	100	96.11	0.32	100	**97.50**	0.35
	Tn = 6	100	97.50	0.37	100	98.11	0.31	100	**98.13**	0.35
	Tn = 7	100	98.33	0.41	100	**99.17**	0.39	100	99.14	0.35
ExYaB	Tn = 10	100	85.60	2.96	100	88.35	1.62	100	**90.12**	2.25
	Tn = 20	100	95.59	4.05	100	96.61	2.76	100	**97.28**	2.87
	Tn = 30	100	97.17	5.67	100	98.82	2.82	100	**98.98**	2.96
UMIST	Tn = 5	100	84.21	0.38	100	87.16	0.24	100	**88.21**	0.26
	Tn = 10	100	96.53	0.70	100	97.60	0.26	100	**98.13**	0.27
	Tn = 15	100	98.18	0.78	100	98.54	0.36	100	**99.27**	0.31

a) The bold number means the best result.

GBLS models achieve better performance than the standard BLS. And for the two GBLS models, IPGBLS can be more effective than the IGBLS in most cases. These differences are even more obvious in recognition with small training sample size. For example, in the UMIST experiment with Tn = 5, the recognition accuracy of three different BLS models is 84.21%, 87.16% and 88.21%, the difference between BLS and GBLS1, GBLS2 is 2.95%, 4.00%. When we set Tn = 15, the difference is only 0.36%, 1.09%. The same phenomenon happens in the other two databases. Therefore, GBLS models may be a better choice to treat the small sample problem (SSP) [27, 28]. Another interesting point is the GBLS models do not consume more running time than standard BLS model, which can be seen from the running time in Table 2. By incorporating a graph regularized term into the objective function of BLS, the new system can attain a better performance with a small amount of nodes (mapping groups, feature nodes and enhancement nodes). Hence, the GBLS model is more effective than the standard BLS. From the experiments presented in this part, we can see that the performance of BLS has indeed been improved by incorporating the graph regularization term. And by considering both the manifold structure and discriminative information of data into the BLS optimization process, the recognition performance can be further improved.

4.2 Comparison with other state-of-the-art classifiers

4.2.1 *Face recognition on clean images*

In this part, we carry out the experiments on the clean ExYaB database over various feature dimensions. Here clean image means an image without occlusion or corruption, just with variations in illumination, expression. The state-of-the-art recognition methods used to be compared with GBLS models includes LRC [29], SVM, NN, SRC [30], CRC_RLS [31] and GELM [25]. The source codes of these competing algorithms are from the authors, and the parameters in each classifier are set to make the algorithm achieve the best performance.

Table 3 The recognition accuracy and running time of competing algorithms on ExYaB[a)]

Method	Dim = 84		Dim = 150		Dim = 300	
	Accuracy (%)	time (s)	Accuracy (%)	time (s)	Accuracy (%)	time (s)
SVM	94.9	5.87	96.4	6.58	97.0	8.26
NN	85.8	3.89	90.0	4.09	91.6	4.76
SRC	95.5	180.90	96.8	205.02	97.9	261.38
LRC	94.5	4.28	95.1	4.72	95.9	6.49
CRC_RLS	95.0	2.12	96.3	2.64	97.9	3.72
GELM	94.45	6.10	95.21	6.90	96.69	7.41
BLS	93.40	0.87	95.05	1.73	96.41	2.25
IGBLS	95.69	0.83	96.90	1.87	98.21	2.31
IPGBLS	**96.11**	1.12	**97.67**	1.91	**98.36**	2.46

a) The bold number means the best result.

For each subject in ExYaB (the detailed characteristics have been showed in Subsection 4.1), we randomly split the images into two parts with equal size, one for training and the other for testing. Following the experimental paradigm in [31], the feature dimension is set as 84, 150 and 300 using Eigenface. Table 3 depicts the recognition results and the running time versus feature dimension by our approaches and other competing classifiers on ExYaB database. It can be found that all the classifiers can attain good recognition results, and the performance can be better with the increase of dimension. And our proposed approaches can achieve a better performance than the competing classifiers consistently, regardless of different dimension setting. Compared with the powerful classifier SRC [30] and CRC_RLS [31], which nearly have reached the saturated point of recognition, the proposed GBLS models can still obtain an improvement of nearly 1 percentage. For the running time, which means the whole time consumed to test all the images in the database, we can see the two GBLS models are faster than the standard BLS and GELM. Compared with the other classifiers, our BLS models (standard BLS, IGBLS and IPGBLS) can attain a better recognition performance with less running time. This illustrates that by incorporating the graph structure, the GBLS models can yield better generalization performance.

4.2.2 *Face recognition with various features*

We further evaluate the performance of our proposed GBLS on a more challenging face database: labeled faces in the wild (LFW) [26]. This database includes 13123 facial images of 5749 subjects captured in real and unconstrained environment with large visual variations in illumination, expression, misalignment, occlusion. Another property of LFW is that the image number of per subject varies from 1 to 530, that means the distribution of data in this database is severely unbalanced. Generally, LFW is utilized to do the face verification. There may be not sufficient samples for certain subjects, the original LFW database cannot be directly applied to do recognition task. To make our proposed GBLS and the other competing algorithms perform well in this database, we need to construct a sub-database of LFW. In this experiment, we select the subset which consists of 143 subjects with more than 11 samples per subject. For each subject, we randomly choose 10 sample images for training and the rest are used for testing.

Instead of using the original unaligned images, three features, e.g., low-frequency Fourier transform feature (FFT), Gabor magnitude and local binary pattern (LBP), are extracted to do the recognition, respectively. For comparison, we consider several competing methods, i.e., SVM, SRC [30], CRC_RLS [31], LRC [29], LCCR [32] and GELM [25]. Table 4 illustrates the testing results versus different feature of LFW. The last column reports the average recognition accuracy of different classifiers on LFW. We can conclude our approaches outperform other competing algorithms. Considering the challenging in LFW, some classifiers, such as SVM and LRC, performs poorly over all the selected features or a certain one. And our IGBLS and IPGBLS methods can perform well consistently. In the Gabor feature case, the IGBLS can perform better than the IPGBLS with a slight advantage, but from the average recognition accuracy, we find the IPGBLS model can achieve the best performance.

Table 4 The recognition accuracy and running time of competing algorithms on the LFW database with three features[a)]

Method	FFT		Gabor		LBP		Average (%)
	Accuracy (%)	time (s)	Accuracy (%)	time (s)	Accuracy (%)	time (s)	
SVM	5.8	9.62	42.4	9.74	18.5	10.96	22.3
SRC	33.6	2230	68.7	2236	61.6	2238	54.6
LRC	13.9	19.35	25.4	20.05	26.3	23.27	21.9
CRC_RLS	14.0	9.11	25.4	9.22	26.3	9.44	21.9
LCCR	22.2	11.49	64.6	11.62	66.5	11.74	51.1
GELM	33.5	5.23	67.8	4.96	58.86	5.68	53.4
BLS	35.2	0.28	69.90	0.33	65.3	0.57	56.8
IGBLS	38.34	0.37	**71.81**	0.35	66.44	0.53	58.86
IPGBLS	**38.85**	0.37	71.32	0.41	**66.76**	0.48	**58.98**

a) The bold number means the best result.

Figure 3 (Color online) Sample images from the four challenging visual database. The first row shows ten kinds of action in Standford 40 Actions database; the second row shows various kinds of birds in CUB200-2011 database; the third row shows ten kinds of flowers in Flower 102 database; the forth row shows ten kinds of objects in Caltech 256 database.

4.2.3 *Recognition on other competitive visual recognition tasks*

The characteristics of the four competitive visual databases are detailed as follows.

• Standford 40 database. This database contains 9352 images from 40 human action, with $180 \sim 300$ images per subject. It is a popular database for action recognition. Following the experimental suggestion in [33], 100 image sample of per class are selected to construct the training set and the rest images for testing.

• Flower 102 database. It includes 8189 flower images from 102 categories with at least 40 images each category. This database attracts much attention of researchers because there exist large variations within the category but smaller difference across several categories.

• CUB200-2011 database. This database, consisting of totally 11788 images of 200 bird species, is popularly used for fine-grained image recognition. It is challenging because of the high degree of similarity among species. The database is split into two halves. One half contains 30 images for per subject for training, and the other half was used for testing.

• Caltech-256 database. This database contains 30608 images of 256 object categories, and each category is consists of at least 80 images. It is commonly applied to evaluate several large-scale image classification algorithms.

A number of images from the above databases are shown in Figure 3.

First, we compare the performance of GBLS models and different competing algorithms on the four challenging visual databases, with the VGG19 features which are extracted by the deep CNN model VGG19 [34]. The feature vectors is with a size of 4096×1, and they are ℓ_2 normalized before doing recognition. The following competing algorithms are chosen for comparison: SVM, kernel SVM, NSC, CRC_RLS, SRC, CROC [35], ProCRC [3] and GELM [25]. The recognition performance on the four

Table 5 Classification performance and running time of different classification on four databases[a)]

Method	Standford 40		Flower 102		CUB200-2011		Caltech 256	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
SVM	79.0	26.97	90.9	30.67	75.4	51.63	80.1	228.52
Kernel SVM	79.8	296.01	92.2	377.24	76.6	691.84	81.3	3085
NSC	74.7	47.16	90.1	67.36	74.5	98.38	80.2	487.27
CRC_RLS	78.2	25.78	93.0	30.74	76.2	49.36	81.1	234.27
SRC	78.7	2655	93.2	3228	76.0	5282	81.3	25535
CROC	79.1	56.32	93.1	74.71	76.2	109.76	81.7	490.53
ProCRC	80.9	26.82	94.8	32.48	78.3	52.27	83.3	234.69
GELM	78.7	54.63	90.3	55.11	76.7	57.75	81.8	69.89
BLS	81.4	14.32	95.3	20.26	78.8	20.74	84.0	26.37
IGBLS	81.7	15.61	95.1	20.33	79.5	21.88	84.6	28.65
IPGBLS	**82.3**	15.92	**95.6**	21.47	**80.4**	23.46	**84.9**	30.42

a) The bold number means the best result.

Table 6 Comparsions to the state-of-the-arts on four challenging visual datasets[a)]

Algorithms	Recognition rate (%)			
	Standford 40	Flower 102	CUB200-2011	Caltech 256
IGBLS	81.7	95.1	79.5	84.6
IPGBLS	**82.3**	**95.6**	**80.4**	84.9
DeepCAMP [36]	52.6	–	–	–
A-FCN [37]	79.7	–	–	–
CNN-SVM [38]	–	74.7	–	–
CNNaug-SVM [38]	–	86.8	66.7	–
FV-CNN [40]	–	–	61.8	–
VGG19 [34]	–	–	–	**85.1**
CNN-S [39]	–	–	–	77.6

a) The bold number means the best result.

databases of all these algorithms are reported in Table 5. From Table 5, some interesting points can be found. Over these four databases, as a simple and naive classifier, NSC always achieves the worst performance. The kernel SVM can perform better than SVM by considering the kernel trick. Three representation-based classifiers: SRC, CROC and CRC_RLS, obtain nearly comparable results with just a slight difference. The ProCRC method derived from the probabilistic view achieve a superior accuracy to those aforementioned algorithms. And our proposed GBLS models can gain an improvement of nearly 1 percentage compared with ProCRC. It can be concluded that the GBLS have the best generalization ability among various kinds of classifiers. Furthermore, we do the comparison between GBLS models (using VGG 19 features) and the state-of-the-art deep neural networks [34, 36–39]. The results are shown in Table 6. Here we should remark that many of the competing networks are CNN-based and the features applied are even better than VGG 19. Compared with these deep networks, the GBLS models obviously attain a significant improvement in recognition. Combining the Tables 5 and 6, we can conclude that our GBLS models possess a significant advantage compared with the shallow and deep methods.

4.3 Parameter selection

In our two proposed GBLS models, there are five common hyper-parameters: three network parameters: N_w, N_f, N_e, and two tradeoff parameters: λ_1, λ_2. In IPGBLS model, there are another two critical parameters, i.e., the number of nearest neighborhood $(k1, k2)$ used to construct the two adjacent graphs. In all our experiments, the three network parameters are selected based on grid search. The selection of the other parameters will be investigated in this part.

We firstly give the parameter selection analysis for IGBLS model. Experiments are carried out on the UMIST database following the same experimental program in Subsection 4.1. The network parameters

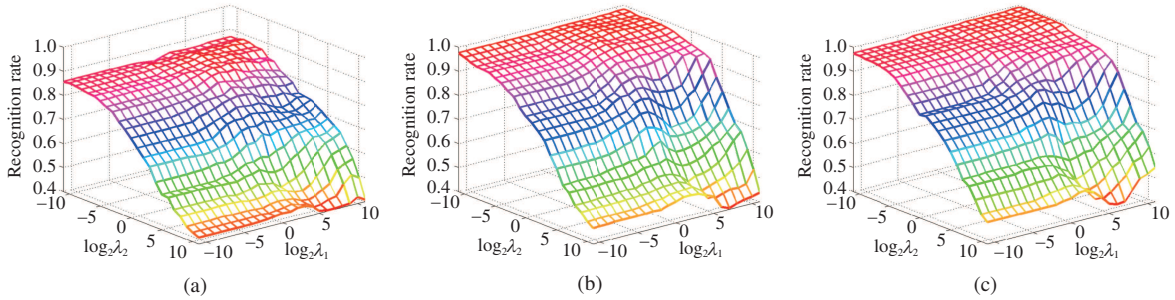

Figure 4 (Color online) Effects of parameters combinations of (λ_1, λ_2) in IGBLS on UMIST. (a) Tn = 5, (b) Tn = 10, (c) Tn = 15.

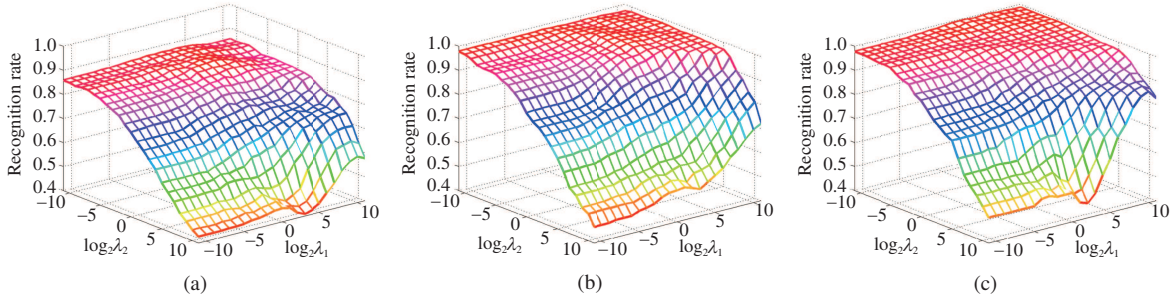

Figure 5 (Color online) Effects of parameters combinations of (λ_1, λ_2) in IPGBLS on UMIST. (a) Tn = 5, (b) Tn = 10, (c) Tn = 15.

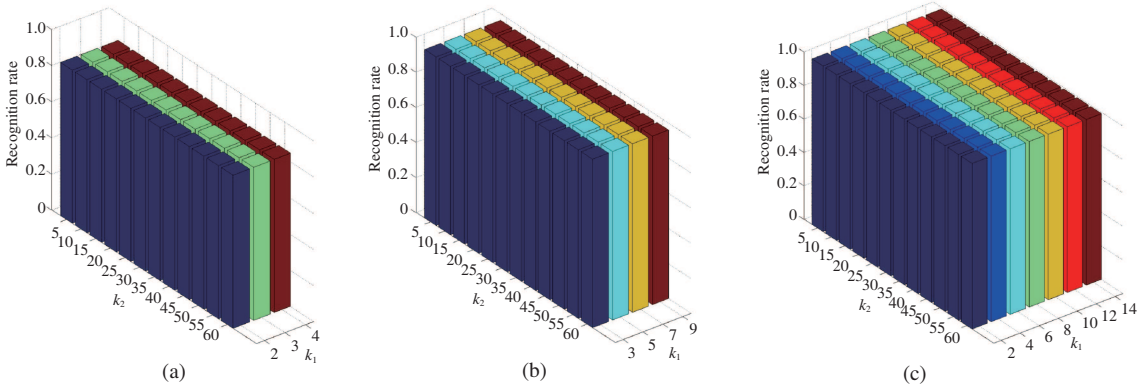

Figure 6 (Color online) Effects of parameters combinations of $(k1, k2)$ in IPGBLS on UMIST. (a) Tn = 5, (b) Tn = 10, (c) Tn = 15.

are fixed as reported in Table 1. We vary λ_1 and λ_2 in candidates $\{2^{-11}, \ldots, 2^{11}\}$. The effects of parameter combination of (λ_1, λ_2) on recognition accuracy is shown in Figure 4. We can find that there exists a large flat area near the optimal point on the landscape over each experimental setting. This means the IGBLS can work well with respect to various combinations of parameters λ_1 and λ_2. From these figures, we can find the optimal combination of (λ_1, λ_2) locates in the upper right part, which means IGBLS model encourages large λ_1 value. This further confirms the necessity of incorporating the label consistency into the objective of BLS.

Secondly, we conduct experiments on UMIST to investigate the parameters selection of IPGBLS model. The tradeoff parameters (λ_1, λ_2) and $(k1, k2)$ are divided into two groups, respectively. For λ_1 and λ_2, the investigation process is just the same as in IGBLS model. Figure 5 shows the experimental results, and we can get the same results as those in IGBLS. For $k1$ and $k2$, we set $k1$ in $\{2, \ldots, \text{Tn}-1\}$ and $k2$ in candidates $\{5, 10, \ldots, 60\}$. Figure 6 shows the results with different combinations of $k1$, $k2$ on different experimental settings. It can be seen that the performance of IPGBLS is very insensitive to the variations

of $k1$ and $k2$. Hence we can choose the appreciate $k1$ and $k2$ in a large range. Thus, we select $k1 = 5$ and $k2 = \mathrm{Tn} - 1$ for all the previous experiments.

5 Conclusion

In this paper, we propose a novel extension of the standard BLS, termed as GBLS, for image recognition. The main contribution of this work is to consider the objective of the standard BLS from the manifold learning perspective. A graph regularization term is incorporated into the objective function, which achieve the minimization of the training error and manifold learning criterion simultaneously. The adjacent matrix can be constructed from two different aspects to form two GBLS models, IGBLS and IPGBLS. The category information is considered into IGBLS to preserve the label consistency of data. While in IPGBLS, it simultaneously considers the geometric structure and discriminative information of training data. Both the GBLS models share the similar objective function which has a closed-form solution, and the resulted output weights can be more discriminative. The experimental results confirm that the GBLS models can possess superior performance for image recognition compared with the standard BLS and several state-of-the-art recognition methods.

This paper has shown the advantage of GBLS in image recognition, we should emphasize that the GBLS can also be applied to other recognition tasks, such as speech recognition and object detection.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant No. 61572540), Macau Science and Technology Development Fund (FDCT) (Grant Nos. 019/2015/A, 024/2015/AMJ, 079/2017/A2), and the University Macau MYR Grants.

References

1 Feng S, Chen C L P. A fuzzy restricted boltzmann machine: novel learning algorithms based on the crisp possibilistic mean value of fuzzy numbers. IEEE Trans Fuzzy Syst, 2018, 26: 117–130
2 Chen C L P, Zhang C Y, Chen L, et al. Fuzzy restricted boltzmann machine for the enhancement of deep learning. IEEE Trans Fuzzy Syst, 2015, 23: 2163–2173
3 Cai S J, Zhang L, Zuo W M, et al. A probabilistic collaborative representation based approach for pattern classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016
4 Xie Z W, Zeng Z, Zhou G Y, et al. Topic enhanced deep structured semantic models for knowledge base question answering. Sci China Inf Sci, 2017, 60: 110103
5 Qu W, Wang D L, Feng S, et al. A novel cross-modal hashing algorithm based on multimodal deep learning. Sci China Inf Sci, 2017, 60: 092104
6 Pao Y H, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities. Computer, 1992, 25: 76–79
7 Pao Y H, Phillips S M, Sobajic D J. Neural-net computing and the intelligent control of systems. Int J Control, 1992, 56: 263–289
8 Igelnik B, Pao Y H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE Trans Neural Netw, 1995, 6: 1320–1329
9 Klassen M, Pao Y H, Chen V. Characteristics of the functional-link net: a higher order delta rule net. In: Proceedings of International Conference on Neural Networks, San Diego, 1988
10 Hornik K. Approximation capabilities of multilayer feedforward networks. Neural Netw, 1991, 4: 251–257
11 Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw, 1989, 2: 359–366
12 Chen C L P. A rapid supervised learning neural network for function interpolation and approximation. IEEE Trans Neural Netw, 1996, 7: 1220–1230
13 Chen C L P, LeClair S R, Pao Y H. An incremental adaptive implementation of functional-link processing for function approximation, time-series prediction, and system identification. Neurocomputing, 1998, 18: 11–31
14 Chen C L P, Wan J Z. A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction. IEEE Trans Syst Man Cybern B, 1999, 29: 62–72
15 Chen C L P, Zhang C Y. Data-intensive applications, challenges, techniques and technologies: a survey on big data. Inf Sci, 2014, 275: 314–347
16 Chen C L P, Liu Z. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE Trans Neural Netw Learn Syst, 2018, 29: 10–24
17 Chen C L P, Liu Z L, Feng S. Universal approximation capability of broad learning system and its structural variations. IEEE Trans Neural Netw Learn Syst, 2018. doi: 10.1109/TNNLS.2018.2866622

18 Feng S, Chen C L P. Fuzzy broad learning system: a novel neuro-fuzzy model for regression and classification. IEEE Trans Cybern, 2018. doi: 10.1109/TCYB.2018.2857815

19 Miao S, Wang J, Gao Q X, et al. Discriminant structure embedding for image recognition. Neurocomputing, 2016, 174: 850–857

20 Fang Y Q, Wang R L, Dai B, et al. Graph-based learning via auto-grouped sparse regularization and kernelized extension. IEEE Trans Knowl Data Eng, 2015, 27: 142–154

21 Yan S C, Xu D, Zhang B Y, et al. Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans Pattern Anal Mach Intel, 2007, 29: 40–51

22 Liao Y Y, Wang Y, Liu Y. Graph regularized auto-encoders for image representation. IEEE Trans Image Process, 2017, 26: 2839–2852

23 Gao Q X, Ma J J, Zhang H L, et al. Stable orthogonal local discriminant embedding for linear dimensionality reduction. IEEE Trans Image Process, 2013, 22: 2521–2531

24 Xue H, Chen S C, Yang Q. Discriminatively regularized least-squares classification. Pattern Recogn, 2009, 42: 93–104

25 Peng Y, Wang S H, Long X Z, et al. Discriminative graph regularized extreme learning machine and its application to face recognition. Neurocomputing, 2015, 149: 340–353

26 Huang G B, Mattar M, Berg T, et al. Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Proceedings of Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, 2008

27 Guo P, Chen C L P, Lyu M R. Cluster number selection for a small set of samples using the Bayesian Ying-Yang model. IEEE Trans Neural Netw, 2002, 13: 757–763

28 Guo P, Lyu M R, Chen C L P. Regularization parameter estimation for feedforward neural networks. IEEE Trans Syst Man Cybern B, 2003, 33: 35–44

29 Naseem I, Togneri R, Bennamoun M. Linear regression for face recognition. IEEE Trans Pattern Anal Mach Intel, 2010, 32: 2106–2112

30 Wright J, Yang A Y, Ganesh A, et al. Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intel, 2009, 31: 210–227

31 Zhang L, Yang M, Feng X C. Sparse representation or collaborative representation: which helps face recognition? In: Proceedings of International Conference on Computer Vision, Barcelona, 2011

32 Peng X, Zhang L, Yi Z, et al. Learning locality-constrained collaborative representation for robust face recognition. Pattern Recogn, 2014, 47: 2794–2806

33 Yao B P, Jiang X Y, Khosla A, et al. Human action recognition by learning bases of action attributes and parts. In: Proceedings of International Conference on Computer Vision, Barcelona, 2011

34 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556

35 Chi Y, Porikli F. Classification and boosting with multiple collaborative representations. IEEE Trans Pattern Anal Mach Intel, 2014, 36: 1519–1531

36 Diba A, Mohammad A, Pirsiavash H, et al. Deepcamp: deep convolutional action & attribute mid-level patterns. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016

37 Wang L M, Qiao Y, Tang X O, et al. Actionness estimation using hybrid fully convolutional networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016

38 Razavian A S, Azizpour H, Sullivan J, et al. CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, 2014

39 Chatfield K, Simonyan K, Vedaldi A, et al. Return of the devil in the details: delving deep into convolutional nets. 2014. ArXiv:1405.3531v4

40 Cimpoi M, Maji S, Vedaldi A. Deep filter banks for texture recognition and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015