Neurocomputing 000 (2018) 1–11

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Kernel group sparse representation classifier via structural and non-convex constraints

Jianwei Zheng[a,b], Hong Qiu[a], Weiguo Sheng[c], Xi Yang[a,*], Hongchuan Yu[b]

[a] School of Computer Science and Technology, Zhejiang University of Technology, 288 Liuhe Road, Hangzhou, Zhejiang, China
[b] National Centre for Computer Animation, Bournemouth University, Bournemouth BH125BB, UK
[c] Institute of Service Engineering, Hangzhou Normal University, 2318 Yuhangtang Road, Hangzhou, Zhejiang, China

ARTICLE INFO

Article history:
Received 17 October 2016
Revised 24 January 2018
Accepted 12 March 2018
Available online xxx

Communicated by Ivor Tsang

Keywords:
Sparse representation
Locality constraint
Group sparse
Kernel trick
Non-convex penalty

ABSTRACT

In this paper, we propose a new classifier named kernel group sparse representation via structural and non-convex constraints (KGSRSN) for image recognition. The new approach integrates both group sparsity and structure locality in the kernel feature space and then penalties a non-convex function to the representation coefficients. On the one hand, by mapping the training samples into the kernel space, the so-called norm normalization problem will be naturally alleviated. On the other hand, an interval for the parameter of penalty function is provided to promote more sparsity without sacrificing the uniqueness of the solution and robustness of convex optimization. Our method is computationally efficient due to the utilization of the Alternating Direction Method of Multipliers (ADMM) and Majorization-Minimization (MM). Experimental results on three real-world benchmark datasets, i.e., AR face database, PIE face database and MNIST handwritten digits database, demonstrate that KGSRSN can achieve more discriminative sparse coefficients, and it outperforms many state-of-the-art approaches for classification with respect to both recognition rates and running time.

1. Introduction

Image Classification (IC) is a fundamental and quintessential task in pattern recognition and computer vision due to its broad applications in human-computer interaction and security monitoring. Exploration to improve upon the accuracy and efficiency of IC approaches has been a remarkable research focus. Deep learning, as one of the popular techniques, has achieved great success due to the facts that it is able to learn extremely powerful hierarchical nonlinear representations of the inputs [1]. However, deep learning based methods require massive training samples, which is difficult to fulfill in many practical applications and computational platforms. Alternatively, the methods via small number of training samples are usually adopted in many specific scenarios.

Recently, sparse representation (SR) has become a very active topic in signal processing and image classification community. So far, it has been successfully applied in many practical problems such as information evaluation [2], feature extraction [3,4], visual tracking [5,6], etc. Under the assumption that natural image can be generally represented by structural primitives, SR-based classifier (SRC) [7] aims to reconstruct a query image using a small set of elements parsimoniously chosen out of an over-complete dictionary, and classifies the query image into the class which results the minimal reconstruction error. In addition, sparse constraints (l_1-norm) not only lead to the unique solution of representation coefficients, but also help to study the actual signal structure. Indeed, most signals admit decomposition over a reduced set of signals from the same class and that will benefit the subsequent classification task. SRC has shown its promise in IC problems and has received remarkable attention.

Some scholars inquired into the reasonability of sparse regularization for IC [8,9]. Zhang et al. [8] argued that it was unnecessary to enforce sparsity constraint on linear regression problem, and asserted it was the collaboration mechanism that makes SRC outperform nearest neighbor based algorithms, such as NN, INNC [10], etc. Correspondingly, they proposed a new classification scheme, namely collaborative representation classifier (CRC). CRC has significantly less computational cost than SRC but leads to very competitive classification results. Both of SRC and CRC follow a reasonable assumption that the subspaces of each objective are independent of each other. However, this assumption is not always held in general image distribution. This may lead to undesirable consequences that some query samples are represented by images from other subjects. To cope with this issue and introduce the

* Corresponding author.
E-mail address: xyang@zjut.edu.cn (X. Yang).

https://doi.org/10.1016/j.neucom.2018.03.035
0925-2312/

essential structure information embedded in the dictionary, Majumdar et al. [11] proposed group sparse classification (GSC) that solves the regression problem in an group way. Under the assumption that the signals can be approximated by a union of a few subspaces, GSC selects certain groups to represent the test sample by using $l_{2,1}$-norm regularization. Similarly, Huang et al. [12] also applied the group sparse coding to images classification, where each test sample is represented by the minimum number of blocks. Both the theoretical analysis and the experimental results have showed the promising performance of GSC, outperforming both SRC and CRC. More recently, it has been verified that the property of locality preservation is more important for a classifier [13,14]. As a result, many regression-based works have been proposed, such as integrating the data locality into the constraints of l_1-norm [15,16], l_2-norm [17,18] or group norm [19,20], for improvement. Moreover, Tang, et al. [21] pointed out that directly involving locality constraint in [19] may disrupt the group structure of sparse coefficients. They further proposed a weighted group sparse representation based on classification (WGSC) by involving the influence of the similarity between query sample and each class.

Despite the successful implementation of regression-based algorithms in IC, there still exist the following limitations: (a) As a result of the linear characteristics, they obtain weak classification result with uniform distribution properties, which is called norm normalization problem in this paper. (b) The rising challenge is to seek for feasible convex regularizations. However, it is well known that non-convex approaches may yield more compact solutions with a fixed residual energy. To deal with the first limitation, many attempts are to introduce different metric representations to fit the underlying structure of samples. Yang et al. [22] proposed a nuclear norm based matrix regression (NMR) for IC, which holds more structural information and performs better in the scenarios of block-corrupted samples. However, it still suffers from the norm normalization problem [23]. Some other works resort to kernel trick to convert linear algorithms into nonlinear forms, such as Kernel SRC (KSRC) [24,25], Kernel CRC (KCRC) [17] and Kernel GSC (KGSC) [13,26,27]. These approaches map the original data into a high-dimensional feature space by using a nonlinear kernel function, and then perform linear optimization in the feature space with the inner products. It has been proved that kernel trick can capture more nonlinear structure of the original data and its performance is better than the linear methods. For the second limitation, non-convex penalty functions are employed, such as the l_p or $l_{2,p}$ pseudo-norm with $p<1$ [28–30].

In this paper, group sparsity with data locality, kernel trick, and the non-convex regularization are further explored, and a joint regression-based approach, named kernel group sparse representation via structural and non-convex constraint (KGSRSN) is proposed. The advantage of integrating all these properties is that more structural information embedded in the dictionary can be captured and then a more discriminative representation can be achieved. Compared to the related works, this paper

(1) Investigates the role of norm normalization step, illuminates the reason for better performance with unit l_2-norm data, and solves this problem by the aid of kernel trick;
(2) presents a formulation of the group sparse coding as a convex optimization problem though defined in terms of non-convex constraint;
(3) derives an efficient iterative approach, using the alternating direction method of multipliers (ADMM) and majorization–minimization (MM), which monotonically decreases the cost function.

The rest of this paper is organized as follows. The previous works are introduced in Section 2. Section 3 explains the reasons why the kernel trick can improve classification performance and then presents our KGSRSN method. Section 4 reports the experiment results on two popular face datasets and the MNIST handwritten dataset. Our conclusions are given in Section 5.

2. Related works

Suppose that we have c classes of subjects, and let $\boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\ldots,\boldsymbol{X}_c]\in\mathbf{R}^{m\times n}$ be the set of training samples, where $\boldsymbol{X}_i=[\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\ldots,\boldsymbol{x}_{in_i}]\in\mathbf{R}^{m\times n_i}$ is the subset of the training samples from subject i, $\mathbf{x}_{ij}$ represents the jth training sample from the ith class, n_i is the number of training samples in class i, $n=\sum_{i=1}^{c}n_i$ is the total sample size and $\boldsymbol{y}\in\mathbf{R}^m$ represents a test sample.

All representation type algorithms have similar principle, i.e. they are premised on over-complete dictionary, all the training samples are distributed in certain subspace. In other words, the test sample $\boldsymbol{y}$ can be represented as a linear combination of $\boldsymbol{X}$

$$\begin{aligned}\boldsymbol{y}&=\boldsymbol{X}_1\boldsymbol{\theta}_1+\boldsymbol{X}_2\boldsymbol{\theta}_2+\cdots+\boldsymbol{X}_c\boldsymbol{\theta}_c\\&=\mathbf{x}_{11}\theta_{11}+\mathbf{x}_{12}\theta_{12}+\cdots+\boldsymbol{x}_{cn_c}\theta_{cn_c}\\&=\boldsymbol{X}\boldsymbol{\theta}\end{aligned}\tag{1}$$

where $\boldsymbol{\theta}=[\theta_{11},\theta_{12},\ldots,\theta_{cn_c}]\in\mathbf{R}^n$ is a coefficient vector corresponding to $\boldsymbol{X}$. Therefore, the sparse solution of Eq. (1) can be recovered as

$$\min_{\theta}||\boldsymbol{\psi}-\boldsymbol{Z}\boldsymbol{\theta}||_2^2+\lambda f(||\boldsymbol{\eta}\odot\boldsymbol{\theta}||_p)\tag{2}$$

where $\odot$ means element-wise multiplication, $\boldsymbol{\eta}\in\mathbf{R}^n$ is the locality weights, $\boldsymbol{\psi}$ and $\boldsymbol{Z}$ are the transformation of the test sample and the training samples, respectively, $f(||\boldsymbol{\theta}||_p)$ leads to the penalty function with a l_p-norm constrained variable, and $\lambda>0$ is a trade-off parameter. Empirically, a larger λ leads to a sparser solution. According to the variation of $\boldsymbol{Z}$, $\boldsymbol{\eta}$ and p, (2) can be turned into different sparse representation algorithms. When (2) is minimized, we obtain the resulting coefficient vector $\boldsymbol{\theta}^*$. Let $\delta_i(\boldsymbol{\theta}^*)$ be a vector whose only nonzero entries are associated with class i. Then the class label of $\boldsymbol{y}$ can be decided as k that gives the minimum reconstruction error, i.e.,

$$k=\arg\min_i||\boldsymbol{\psi}-\boldsymbol{Z}\delta_i(\boldsymbol{\theta}^*)||_2\tag{3}$$

2.1. Linear representation classification

The classical SR-based approaches are developed with different setting of penalty functions f and norm constraints p, by utilizing $\boldsymbol{\psi}=\boldsymbol{y},\boldsymbol{Z}=\boldsymbol{X}$ directly and having no consideration of weights constraint ($\boldsymbol{\eta}=\mathbf{1}_n$).

For f being the absolute function and p=1, (2) turns to the classical SRC [7]. Its coding coefficients $\boldsymbol{\theta}$ is sparse under l_1-norm constraint and over-complete dictionary. Suppose the test sample $\boldsymbol{y}$ belongs to the ith subject, then all the coding coefficients will shrink to be zero except $\boldsymbol{\theta}_i$.

For f being the square function and p=2, then (2) becomes the classical CRC [8]. While the importance of sparsity is much emphasized in SRC, Zhang et, al. [8] argued that the success of SRC is attributed to collaborative mechanism rather than sparsity. Furthermore, CRC has significantly less complexity than SRC by using l_2-minimization. It is noteworthy that CRC is not sparse since its coding coefficient will not tend to absolute zero.

SRC and CRC are all unsupervised learning algorithms, they ignore the label information during model establishment. GSC selects a few groups to represent the query sample by using $l_{2,1}$-norm regularizer. Setting p=2, 1 and still with the absolute function f, it uses l_1-norm in inter-class samples while using l_2-norm in intra-class samples. Previous studies indicate that the recognition rate of GSC is superior to that of SRC and CRC [12], but its efficiency is inferior to the closed solution of CRC.

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

2.2. Weighted representation classification

As described above, SRC, CRC and GSC construct the classification model respectively by sparsity, collaboration and supervision. They ignore the locality structure of the training samples. Recently, scholars have proposed many locality preserving methods. Similar to the classical ones, weighted representation approaches involve the considerations of $\boldsymbol{\eta}$ while keeping other terms consistently. $\boldsymbol{\eta}$ measures the Euclidean distance between the test samples and the training samples as

$$\eta_{ij} = \exp(\| \boldsymbol{x}_{ij} - \boldsymbol{y} \|_2^2 / \sigma^2) \tag{4}$$

where η_{ij} denotes the jth weights from the ith class, and σ is a scalar parameter.

The weighted extensions of SRC, CRC and GSC are weighted SRC (WSRC) [15,16], weighted CRC (WCRC) [17,18], and locality group sparse representation (LGSR) [19], respectively. These approaches make the coding coefficient of remote samples shrink to zero while neighbor samples obtain larger coding coefficient, which means that they have the properties of locality and noise-resistance. In addition, WGSC [21] turns the regularization term of (2) to be $\sum_{i=1}^{c} r_i ||\boldsymbol{\eta}_i \odot \boldsymbol{\theta}_i||_2$, which further include the weight factor r_i for better holding the group structure.

2.3. Kernel representation classification

Kernel-based methods adopt the kernel trick to map the original data into a high-dimensional feature space by using an implicit nonlinear mapping function, and then perform linear processing in this high-dimensional space with inner products. Particularly, as for SRC, CRC, and GSC, their kernel versions hold the same regularization term but with $\boldsymbol{y}$ and $\mathbf{X}$ be implicitly mapped.

In the kernel space, $\boldsymbol{\psi}$ and $\boldsymbol{Z}$ are expressed as $\phi(\boldsymbol{y})$ and $\Phi(\boldsymbol{X})$, respectively, where $\phi(\cdot)$ denotes the mapping function and Φ represents its matrix form. Thus, (2) can be rewritten as

$$\min_{\theta} \| \boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\theta} \|_2^2 + \lambda f(\| \boldsymbol{\theta} \|_p) \tag{5}$$

where the weight constraint is temporarily ignored, i.e. set $\boldsymbol{\eta} = \mathbf{1}_n$. Since ϕ is unknown, (5) can be reformulated by some kernel functions as

$$\begin{aligned} &||\phi(\boldsymbol{y}) - \Phi(\boldsymbol{X})\boldsymbol{\theta}||_2^2 + \lambda f(||\boldsymbol{\theta}||_p) \\ &= (\phi(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\theta}) + \lambda f(||\boldsymbol{\theta}||_p) \\ &= k(\boldsymbol{y},\boldsymbol{y}) + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\theta} - 2k(\bullet,\boldsymbol{y})^{\mathrm{T}}\boldsymbol{\theta} + \lambda f(||\boldsymbol{\theta}||_p) \end{aligned} \tag{6}$$

where $\boldsymbol{K} = \boldsymbol{\Phi}(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{X}) \in \mathbf{R}^{n\times n}$ is a symmetric positive semi-definite kernel matrix. $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the given kernel function, and $k(\bullet,\boldsymbol{y}) = [k(\boldsymbol{x}_1,\boldsymbol{y}),\ldots,k(\boldsymbol{x}_n,\boldsymbol{y})] = \Phi(\boldsymbol{X})^{\mathrm{T}}\phi(\boldsymbol{y})$. There are some commonly used kernel functions, i.e., the linear kernel, polynomial kernel as well as the most frequently used Gaussian kernel that is defined as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 / 2\sigma^2) \tag{7}$$

where σ is a tunable parameter.

Integrating (7) and (4) into (2) can lead to kernel weighted version of SR-based approaches, such as Kernel WCRC (KWCRC) [17]. Literature [17] and [24] demonstrate that the performance of kernel weighted representation classification is superior to that of other representation type classification empirically.

3. Kernel group sparse representation classifier via structural and non-convex constraints

To improve the weighted group constraint and further enforce it in the kernel space, a new non-convex penalty function and the kernel trick is adopted to present a joint-sparsity SR-based method, namely kernel group sparse representation classifier via structural and non-convex constraints (KGSRSN). In KGSRSN, the locality metrics are measured in the kernel space, thus the nonlinear structure of input samples would be better explored. In this section, we first explain the relationship between data normalization and sample selection, and then present the kernel-based method to deal with the norm normalization problem. Second, we chose a specific non-convex constraint in parametric form, so that the complete objective function would be strictly convex. Finally, we derive the optimization algorithm according to the principle of ADMM and MM.

3.1. The norm normalization problem

In the practical IC problems, some normalization steps, such as mean normalization [31] or norm normalization [23], always conducted in advance. These steps can regulate the distribution of input data and enhance the numerical stability of classifiers, but the consequence has not been analyzed in theory and empirically. By a concrete example, we investigate the impact of norm normalization on sparse representation algorithms in this section.

Consider the samples in Fig. 1(a) as an example, where the dataset consists of 150 points from three classes. all of them are given by standard Gaussian distribution and without any normalization. We choose the red dot (1, 2) as the test sample and the others as training samples. From Fig. 1(b) and 1(c), it can be seen that the selected samples for representing the red dot mainly involves the ones from different classes, which indicates that the sparse representation has no discriminality in this case. We call this phenomenon as the norm normalization problem. The reason is that the data points in the data set may have different l_2-norm (or l_1-norm), so the sparse representation of one point may be inclined to select the data with larger l_2-norm if possible. For this dataset, the l_2-norm of round samples, square samples and star samples, respectively range from 0.433 to 17.668, 13.194 to 48.659 and 3.494 to 31.602. It is concluded that the l_2-norm of square samples and star samples is significantly higher than that of round samples. Thus the sparse representation of the red dot may be inclined to select the square points and star points. In Fig. 1(b), the selected samples with nonzero coefficients of SRC are marked with one blue square and one blue star, not the right round data. It violates the basic idea of SRC that is to represent a query sample as an intra-class sparse linear combination of the training samples. For CRC and GSC, we choose the 15% largest elements of their representation coefficients for output, since they are not sparse in strict significance. Although the effective samples of CRC involve three different kinds of input data, the majority is the square and star points in Fig. 1(c). In Fig. 1(d), GSC eliminates square samples by group constraint, but the number of star points is still much more than that of round points. In Fig. 1(e), WSRC represents the query sample as an intra-class sparse linear combination of the training samples benefiting from the property of locality. However, the selected representation samples are relatively with larger l_2-norms, the norm normalization problem still exists.

To illustrate the importance of l_2-norm, we further take the two-dimensional case as an example. Suppose that there are two classes $\boldsymbol{X}_1 = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$ and $\boldsymbol{X}_2 = [\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6]$ which are normalized to have unit l_2-norm and distributed on a unit sphere in Fig. 2. Here $\boldsymbol{x}_1$ is regarded as a test sample while the rest as training samples. Denote $\boldsymbol{q}_1$ as the junction of the line linking $\boldsymbol{x}_2$ to $\boldsymbol{x}_3$ with the line linking $\boldsymbol{x}_1$ to the origin, and denote b_1 as the Euclidean distance from $\boldsymbol{q}_1$ to the origin. Similarly, denote $\boldsymbol{q}_2$ as the junction of the line linking $\boldsymbol{x}_2$ to $\boldsymbol{x}_4$ with the line linking $\boldsymbol{x}_1$ to the origin, and denote b_2 as the distance from $\boldsymbol{q}_2$ to the origin. Assume that $\boldsymbol{x}_1$ can be represented as a linear combination of $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$. Then

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

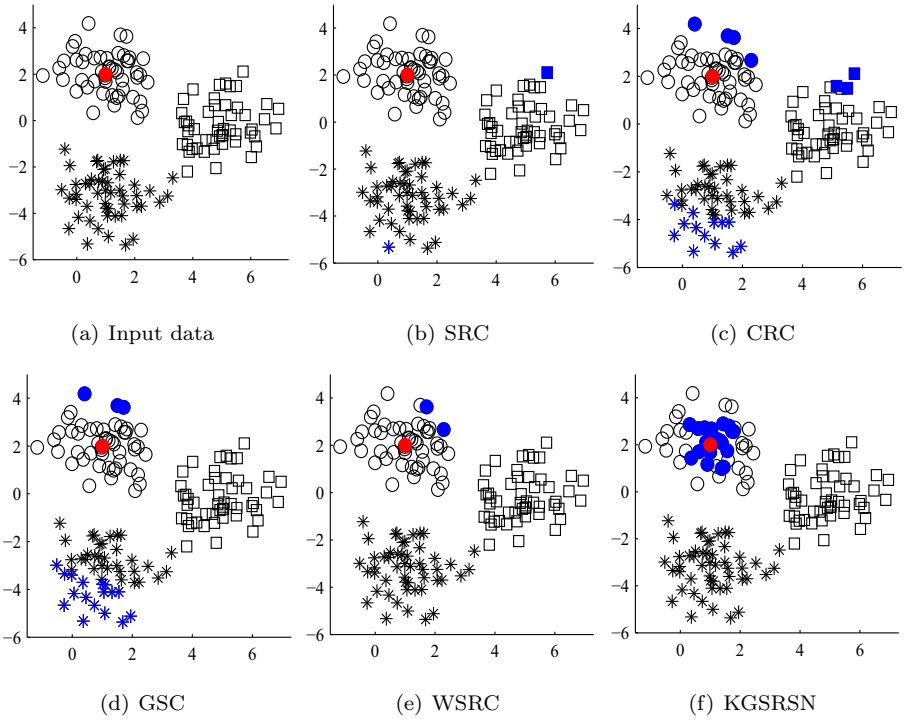


Fig. 1. Representation results of some non-normalized data from typical SR-based methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

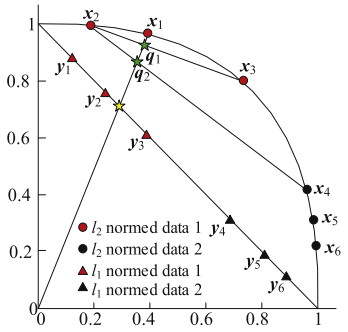


Fig. 2. The illustration of norm normalization problem.

$\boldsymbol{x}_1$ can be denoted as

$$\begin{aligned}\boldsymbol{x}_1 &= \frac{1}{b_1}(\boldsymbol{q}_1) = \frac{1}{b_1}(\theta_2\boldsymbol{x}_2 + \theta_3\boldsymbol{x}_3)\\ &= [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_6][0, \frac{\theta_2}{b_1}, \frac{\theta_3}{b_1}, \ldots, 0]^{\mathrm{T}}\end{aligned}$$

where θ_2 and θ_3 are some positive numbers satisfying $\theta_2 + \theta_3 = 1$. For SRC, $\boldsymbol{x}_1 = [\boldsymbol{X}_1\boldsymbol{X}_2]\boldsymbol{\theta}$ and $||\boldsymbol{\theta}||_1 = 1/b_1$. Similarly, if $\boldsymbol{x}_1$ can be represented as a linear combination of $\boldsymbol{x}_2$ and $\boldsymbol{x}_4$ by $\boldsymbol{\theta}' = [0, \theta_2'/b_2, 0, \theta_2'/b_2, 0, 0]^{\mathrm{T}}$, then $\boldsymbol{x}_1 = [\boldsymbol{X}_1\boldsymbol{X}_2]\boldsymbol{\theta}'$ and $||\boldsymbol{\theta}'||_1 = 1/b_2$. From Fig. 2, it can be observed that $b_1 > b_2$, hence $||\boldsymbol{\theta}||_1 < \boldsymbol{\theta}'||_1$, which makes SRC select $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ as the representation samples. In other words, when all samples have the same l_2-norm, the linear representation of a test data will be declined to select neighbor points from the same class, which leads to the property of discriminability under the well-known assumption that intra-class samples always agglomerated closer than inter-class samples. Furthermore, triangle data in Fig. 2 are normalized to have unit l_1-norm. It is obvious that this pretreatment can also regulate the structure of input data, but the output data all lie on one line [32], which makes q_1 and q_2 overlap to a single point (the yellow star), i.e., b_1=b_2. Therefore, l_1-norm has less discriminality than l_2-norm.

3.2. The proposed method

For any data point $\boldsymbol{x}$, we have $||\boldsymbol{\phi}(\boldsymbol{x})||_2^2 = k(\boldsymbol{x}, \boldsymbol{x}) = 1$ from (7), so the data point $\phi(\boldsymbol{x})$ naturally have unit l_2-norm. Since kernel trick can make the data points in high-dimensional feature space linearly separable, the kernel-based sparse representation of data can be a reasonable strategy to solve the norm normalization problem. From Fig. 1(f), we can see that KGSRSN obtains the accurate representation data that distributes around the test samples.

For the penalty function on the other hand, many works involved l_p-norm regularizer for more compact representation. However, this non-convex constraint generally suffers from many numerical problems such as suboptimal local solutions, slow convergence rate, and some initialization issues. In this section, we introduce a new penalty function for more accurate data reconstruction, while maintaining convexity of the total cost function. For this purpose, the logarithmic penalty [33],

$$f(x) = \frac{\log(1 + a|x|)}{a} \tag{8}$$

is adopted, where $a > 0$ is a scalar parameter.

Integrate (2), (7), and (8) into WGSC, the cost function of our KGSRSN is

$$\min_{\theta} \frac{1}{2}||\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\theta}||_2^2 + \lambda \sum_{i=1}^{c} r_i \frac{\log(1 + a||\boldsymbol{d}_i \odot \boldsymbol{\theta}_i||_2)}{a} \tag{9}$$

where $\boldsymbol{d}_i$ is locality metric, whose entry d_{ij} measures the distance between $\boldsymbol{y}$ and training sample $\boldsymbol{x}_{ij}$ in the kernel space as

$$\begin{aligned}d_{ij} =& ||\boldsymbol{\phi}(\boldsymbol{x}_{ij}) - \phi(\boldsymbol{y})||_2^2\\ =& \phi(\boldsymbol{x}_{ij})^{\mathrm{T}}\phi(\boldsymbol{x}_{ij}) - 2\phi(\boldsymbol{x}_{ij})^{\mathrm{T}}\phi(\boldsymbol{y}) + \phi(\boldsymbol{y})^{\mathrm{T}}\phi(\boldsymbol{y})\\ =& 2 - 2k(\boldsymbol{x}_{ij}, \boldsymbol{y})\end{aligned} \tag{10}$$

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

r_i is the group weight to indicate how well $\mathbf{X}_i$ can represent $\mathbf{y}$. A smaller r_i denotes larger probability of $\mathbf{y}$ belonging to the ith class. By learning from the idea of LRC [34], r_i can be defined as

$$\begin{aligned} r_i =& ||\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X}_i)\boldsymbol{\theta}_i^*||_2^2 \\ =& 1 - 2\boldsymbol{k}_i(\bullet,\boldsymbol{y})^{\mathrm{T}}\boldsymbol{\theta}_i^* + \boldsymbol{\theta}_i^{*\mathrm{T}}\boldsymbol{K}_i\boldsymbol{\theta}_i^* \\ \boldsymbol{\theta}_i^* =& \arg\min ||\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X}_i)\boldsymbol{\theta}_i||_2^2 = \boldsymbol{K}_i^{-1}\boldsymbol{k}_i(\bullet,\boldsymbol{y}) \end{aligned} \tag{11}$$

where $\boldsymbol{K}_i = \boldsymbol{\Phi}(\boldsymbol{X}_i)^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{X}_i) \in \mathbf{R}^{n_i \times n_i}$ is the ith kernel matrix, $k_i(\bullet,\boldsymbol{y}) = \boldsymbol{\Phi}(\boldsymbol{X}_i)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{y})$ is the kernel vector between $\boldsymbol{y}$ and the training samples from the ith class.

Similar as (3), the final decision rule can be formulated as

$$\begin{aligned} k &= \arg\min_i ||\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X}_i)\delta_i(\boldsymbol{\theta}^*)||_2 \\ &= 1 - 2\boldsymbol{k}_i(\bullet,\boldsymbol{y})^{\mathrm{T}}\boldsymbol{\theta}_i^* + \boldsymbol{\theta}_i^{*\mathrm{T}}\boldsymbol{K}_i\boldsymbol{\theta}_i^* \end{aligned} \tag{12}$$

i.e., the class label of $\boldsymbol{y}$ is decided as the class which gives the minimum residual error in the kernel feature space.

Notice that the chosen kernel needs to satisfy $k(\boldsymbol{x},\boldsymbol{y}) = c$ for avoiding norm normalization problem, where c is a constant. This can be satisfied when it is an isotropic kernel $k(\boldsymbol{x},\boldsymbol{y}) = k(\|\boldsymbol{x},\boldsymbol{y}\|)$ or any normalized kernel $k(\boldsymbol{x},\boldsymbol{y}) = k(\boldsymbol{x},\boldsymbol{y})/(k(\boldsymbol{x},\boldsymbol{x})^{1/2} \times k(\boldsymbol{y},\boldsymbol{y})^{1/2})$. In this paper, we adopt the Gaussian kernel in our experiments.

3.3. Optimization algorithm

Although some well-known algorithms, such as Homotopy [35] and sparse projections [36], have been deliberatively developed for SR-based approaches, they cannot be used directly for our method since it integrates both locality group sparsity and non-convex penalty in the kernel space. We derive our optimization algorithm jointly under the framework of ADMM [37,38] and MM [39,40]. The former is efficient for most convex coding problems, and the later replaces some tough problems by simpler ones. For convenience, we introduce matrix $\mathbf{D}$=diag($[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_c]$) $\in \mathbf{R}^{n\times n}$ and define $\boldsymbol{\beta} = \boldsymbol{D}\boldsymbol{\theta}$. Then our KGSRSN algorithm can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{\phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{D}^{-1}\boldsymbol{\beta}||_2^2 + \lambda\sum_{i=1}^{c} r_i \frac{\log(1+a||\boldsymbol{\beta}_i||_2)}{a} \\ &= \min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\tilde{\boldsymbol{K}}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{T}}\tilde{k}(\bullet,\boldsymbol{y}) + \lambda\sum_{i=1}^{c} r_i \frac{\log(1+a||\boldsymbol{\beta}_i||_2)}{a} \end{aligned} \tag{13}$$

To solve (13), we first define the augmented Lagrangian function as

$$\begin{aligned} L_\rho(\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{\mu}) =& \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\tilde{\boldsymbol{K}}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{T}}\tilde{k}(\bullet,\boldsymbol{y}) + \lambda\sum_{i=1}^{c} r_i \frac{\log(1+a||\boldsymbol{z}_i||_2)}{a} \\ &+ \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{\beta}-\boldsymbol{z}\|_2^2 \end{aligned} \tag{14}$$

where $\boldsymbol{\mu}$ is the Lagrange multiplier and $\rho > 0$ is a penalty parameter. Solving (14) is equivalent to tracing the solutions $(\boldsymbol{\beta}^*, \boldsymbol{z}^*, \boldsymbol{\mu}^*)$ of the saddle-point problem [38]:

$$L(\boldsymbol{\beta}^*,\boldsymbol{z}^*,\boldsymbol{\mu}) \le L(\boldsymbol{\beta}^*,\boldsymbol{z}^*,\boldsymbol{\mu}^*) \le L(\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{\mu}^*) \tag{15}$$

With the computed (or initialized) vector $\boldsymbol{z}^k$ and $\boldsymbol{\mu}^k$, by applying the ADMM iterative scheme to (15), the update of each variable goes as

$$\boldsymbol{\beta}^{k+1} \leftarrow \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta},\boldsymbol{z}^k,\boldsymbol{\mu}^k) \tag{16a}$$

$$\boldsymbol{z}^{k+1} \leftarrow \arg\min_{\boldsymbol{z}} L(\boldsymbol{\beta}^{k+1},\boldsymbol{z},\boldsymbol{\mu}^k) \tag{16b}$$

$$\boldsymbol{\mu}^{k+1} \leftarrow \boldsymbol{\mu}^k + \rho(\boldsymbol{\beta}^{k+1} - \boldsymbol{z}^{k+1}) \tag{16c}$$

Fixing $\boldsymbol{z}^k$ and $\boldsymbol{\mu}^k$, the subproblem for $\boldsymbol{\beta}^{k+1}$ can be rewritten as

$$\boldsymbol{\beta}^{k+1} \leftarrow \arg\min_{\boldsymbol{\beta}} \left(\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\tilde{\boldsymbol{K}}\boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{T}}\tilde{k}(\bullet,\boldsymbol{y}) + \frac{\rho}{2}\|\boldsymbol{\beta} - \boldsymbol{z}^k + \frac{\boldsymbol{\mu}^k}{\rho}\|_2^2\right) \tag{17}$$

where the constant terms have been omitted. The first-order optimality conditions of quadratic minimization problem (17) lead to

$$\boldsymbol{\beta}^{k+1} = (\tilde{\boldsymbol{K}} + \rho\boldsymbol{I})^{-1}(\tilde{k}(\bullet,\boldsymbol{y}) + \rho\boldsymbol{z}^k - \boldsymbol{\mu}^k) \tag{18}$$

Fixing $\boldsymbol{\beta}^{k+1}$ and $\boldsymbol{\mu}^k$, the update of $\boldsymbol{z}^{k+1}$ is given by minimizing the following subproblem:

$$\boldsymbol{z}^{k+1} \leftarrow \arg\min_{\boldsymbol{z}} \left(\frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{\beta}^{k+1} - \frac{\boldsymbol{\mu}^k}{\rho}\|_2^2 + \lambda\sum_{i=1}^{c} r_i \frac{\log(1+a||\boldsymbol{z}_i||_2)}{a}\right) \tag{19}$$

Solving subproblem (19) is difficult due to the involving of group constraint and non-convex logarithmic penalty. We use the MM procedure to derive a method minimizing (19). Firstly, Proposition 1 is presented to specify a majorizer of the logarithmic function.

Proposition 1. *The function q defined by*

$$q(x,v) = \frac{x^2}{2v(1+av)} + \frac{\log(1+av)}{a} - \frac{v}{2(1+av)} \tag{20}$$

is a majorizer of the logarithmic function except for v=0, i.e.,

$$q(x,v) \ge \frac{\log(1+ax)}{a}, \forall x \in R, v \in R\backslash\{0\} \tag{21a}$$

$$q(v,v) = \frac{\log(1+av)}{a}, \forall v \in R\backslash\{0\} \tag{21b}$$

Proof. (21b) can be verified by a simple substitution. For (21a), using Talors expansion, we get

$$\frac{\log(1+ax)}{a} = \frac{\log(1+av)}{a} + \frac{(x-v)}{(1+av)} - \frac{a(x-v)^2}{2(1+av_0)^2} \tag{22}$$

for v_0 between v and x. Since $a > 0$, we have

$$\frac{\log(1+ax)}{a} \le \frac{\log(1+av)}{a} + \frac{(x-v)}{(1+av)} \tag{23}$$

Using $x \le x^2/2v + v/2$, we further get

$$\frac{\log(1+ax)}{a} \le \frac{x^2}{2v(1+av)} + \frac{\log(1+av)}{a} - \frac{v}{2(1+av)} = q(x,v) \tag{24}$$

which completes the proof. □

From Proposition 1, the total function

$$\begin{aligned} Q(\boldsymbol{z},\boldsymbol{z}^k) &= \frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{\beta}^{k+1} - \frac{\boldsymbol{\mu}^k}{\rho}\|_2^2 + \frac{\lambda}{2}\sum_{i=1}^{c} r_i \frac{\|\boldsymbol{z}_i\|_2^2}{\|\boldsymbol{z}_i^k\|_2(1+a\|\boldsymbol{z}_i^k\|_2)} + C \\ &= \sum_{i=1}^{c}\left(\frac{\rho}{2}\|\boldsymbol{z}_i - \boldsymbol{\beta}_i^{k+1} - \frac{\boldsymbol{\mu}_i^k}{\rho}\|_2^2 + \frac{\lambda r_i\|\boldsymbol{z}_i\|_2^2}{2\|\boldsymbol{z}_i^k\|_2(1+a\|\boldsymbol{z}_i^k\|_2)}\right) + C \end{aligned} \tag{25}$$

is a majorization surrogate function of subproblem (19) with C being a constant term independent of $\boldsymbol{z}$. We then can obtain all $\boldsymbol{z}_i^{k+1}, i = 1, \ldots, c$, as follows:

$$\boldsymbol{z}_i^{k+1} = \left(\rho + \frac{\lambda r_i}{\|\boldsymbol{z}_i^k\|_2(1+a\|\boldsymbol{z}_i^k\|_2)}\right)^{-1}\left(\rho\boldsymbol{\beta}_i^{k+1} + \boldsymbol{\mu}_i^k\right) \tag{26}$$

The details of our KGSRSN are given in Algorithm 1.

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

Algorithm 1 KGSRSN.

Input: **D**, **K**, $k(\bullet, \boldsymbol{y})$, $\mathbf{r}$, λ, a, ρ, ε_1, and ε_2.
Initialize $\boldsymbol{z} = \boldsymbol{\mu} = \mathbf{0}$, $i_o = i_i = 0$.
Repeat
1. $i_o = i_o + 1$.
2. Update $\boldsymbol{\beta}$ by (18)
   Repeat
   3. Initialize $\mathbf{z} = \boldsymbol{\beta} + \boldsymbol{\mu}/\rho$.
   4. $i_i = i_i + 1$.
   5. Update $\mathbf{z}_j$, $j = 1, 2, \ldots, c$, by (26).
   Until the value of subploblem (19) satisfies that (norm(cost(i_i)-cost($i_i - 1$))/norm(cost(i_i)) $< \varepsilon_2$
6. Update $\boldsymbol{\mu}$ by (16c).
Until norm($\boldsymbol{\beta} - \boldsymbol{z}$) $< \varepsilon_1$
Output: $\boldsymbol{\theta}^* = \boldsymbol{D}^{-1}\boldsymbol{\beta}$.

3.4. Convergence and complexity analysis

The convergence properties of ADMM have been extensively studied [22,37,38]. Following the ideas of Ref. [38], Proposition 2 holds under the assumption that the functions of (16a) and (16b) are closed, proper, and convex.

Proposition 2 ([38]). *The ADMM iterates satisfy that*

(1) *Residual convergence.* $\boldsymbol{\beta}^k - \boldsymbol{z}^k \to 0$ *as* $k \to \infty$, *i.e., the iterates approach feasibility.*
(2) *Objective convergence. The cost function (13) of the iterates approaches the optimal value.*
(3) *Dual variable convergence.* $\boldsymbol{\mu}^k \to \boldsymbol{\mu}^*$ *as* $k \to \infty$, *where* $\boldsymbol{\mu}^*$ *is a dual optimal point.*

With the fact that the closedness and properness of our cost function (13) are clear, we further need to make sure that all the subproblems be convex and have convergent solutions. It is evident that subproblem (17) is strictly convex and with closed-form solution (18), the remaining issue for us is to prove the convexity and convergence of subproblem (19). Above all, we introduce Proposition 3 to assert the condition for convexity.

Proposition 3. *Subproblem (19) is strictly convex under the condition that*

$$0 < a < \frac{\rho}{\lambda \max(r_i)} \tag{27}$$

Proof. For any $v \geq 0$, define function g: $R \to R$ with parameter $a > 0$ as

$$g(v) = \frac{\rho}{2}v^2 + \lambda r f(v) \tag{28}$$

Since the logarithmic function $f(v)$ is continuous and twice differentiable for $v \geq 0$, the convexity of g can be ensured by a positive second derivative on $v \geq 0$. This leads to the condition

$$g''(v) = \rho + \lambda r f''(v) > 0 \Rightarrow f''(v) > -\rho/(\lambda r) \tag{29}$$

Moreover, since $f''(v) = -a/(1+av)^2$, and the minimum second-order derivative of $f(v)$ resides in $f''(0) = -a$, we can obtain that

$$-a > -\rho/(\lambda r) \Rightarrow a < \rho/(\lambda r) \tag{30}$$

Based on the convexity of $\|\boldsymbol{x}\|_2$, $g(\|\boldsymbol{x}\|_2)$ is also strictly convex. By expanding and decomposing (19) into

$$\begin{aligned}
&\frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{\beta}^{k+1} - \frac{\boldsymbol{\mu}^k}{\rho}\|_2^2 + \lambda \sum_{i=1}^{c} r_i \frac{\log(1 + a\|\boldsymbol{z}_i\|_2)}{a} \\
&= \frac{\rho}{2}\|\boldsymbol{z}\|_2^2 - \frac{\rho}{2}\boldsymbol{z}^{\mathrm{T}}\left(\boldsymbol{\beta}^{k+1} + \frac{\boldsymbol{\mu}^k}{\rho}\right) + \lambda \sum_{i=1}^{c} r_i \frac{\log(1 + a\|\boldsymbol{z}_i\|_2)}{a} + C \\
&= \sum_{i=1}^{c} g(\|\boldsymbol{z}_i\|_2) - \frac{\rho}{2}\boldsymbol{z}^{\mathrm{T}}\left(\boldsymbol{\beta}^{k+1} + \frac{\boldsymbol{\mu}^k}{\rho}\right) + C,
\end{aligned} \tag{31}$$

we can see that it is a linear combination of $g(\|\boldsymbol{z}_i\|_2)$, a convex term $\boldsymbol{z}^{\mathrm{T}}(\boldsymbol{\beta}^{k+1} + \boldsymbol{\mu}^k/\rho)$, and a constant term C. Hence, (19) is strictly convex with condition (27). □

To analyze the convergence of $\boldsymbol{z}$ subproblem, we first characterize a δ strongly convexity of the surrogate function $Q(\boldsymbol{z}, \boldsymbol{z}^k)$ in (25) as Proposition 4. Then by the important property shown in Lemma 1 [40], we give convergence results for subproblem $\boldsymbol{z}$ as in Proposition 5.

Proposition 4. *The surrogate function* $Q(\boldsymbol{z}, \boldsymbol{z}^k)$ *in (25) is* δ *strongly convex, i.e., function* $Q(\boldsymbol{z}, \boldsymbol{z}^k) - 0.5\delta||\boldsymbol{z}||_2^2$ *is convex, where* $\delta > 0$.

The proof is omitted here since its derivation is very similar as the proof of Proposition 3.

Lemma 1 [40]. *Let* $J(\boldsymbol{x})$: $R^n \to R$ *be a* δ *strongly convex function, and* $\boldsymbol{x}^*$ *be the minimizer of* $J(\boldsymbol{x})$, *then inequality* $0.5\delta||\boldsymbol{x} - \boldsymbol{x}^*||_2^2 \leq J(\boldsymbol{x}) - J(\boldsymbol{x}^*)$ *holds for any* $\boldsymbol{x} \in R^n$.

Proposition 5. *Denote* $J(\boldsymbol{z})$ *as subproblem (19), let* $\{\boldsymbol{z}^k\}_{k=1}^{\infty}$ *be the generated sequence by the inner loop of Algorithm 1 with any initial* $\boldsymbol{z}^0$, *then we have:*

(1) *The sequence* $\{J(\boldsymbol{z}^k)\}_{k=0}^{\infty}$ *is monotonically non-increasing and convergent;*
(2) *The property* $\lim_{k\to\infty}||\boldsymbol{z}^k - \boldsymbol{z}^{k+1}||_2^2 = 0$ *holds for the corresponding sequence* $\{\boldsymbol{z}^k\}_{k=0}^{\infty}$.

Proof. Recall Proposition 4, the surrogate function $Q(\boldsymbol{z}, \boldsymbol{z}^k)$ is a δ strongly convex majorizer of $J(\boldsymbol{z})$ at $\boldsymbol{z}^k$, and $\boldsymbol{z}^{k+1}$ is the global minimizer of $Q(\boldsymbol{z}, \boldsymbol{z}^k)$. Thus for any $k \geq 0$ we can write:

$$J(\boldsymbol{z}^{k+1}) \leq Q(\boldsymbol{z}^{k+1}, \boldsymbol{z}^k) \leq Q(\boldsymbol{z}^k, \boldsymbol{z}^k) = J(\boldsymbol{z}^k) \tag{32}$$

which reveals the monotonically non-increasing property of sequence $\{J(\boldsymbol{z}^k)\}_{k=0}^{\infty}$. Notice that $\{J(\boldsymbol{z}^k)\}_{k=0}^{\infty}$ is bounded from below with zero, hence convergent.

Lemma 1 with $Q(\boldsymbol{z}, \boldsymbol{z}^k)$ in place of $J(\boldsymbol{x})$ and $\boldsymbol{z}^{k+1}$ in place of $\boldsymbol{x}^*$ yields

$$\frac{\delta}{2}||\boldsymbol{z} - \boldsymbol{z}^{k+1}||_2^2 \leq Q(\boldsymbol{z}, \boldsymbol{z}^k) - Q(\boldsymbol{z}^{k+1}, \boldsymbol{z}^k), \forall \boldsymbol{z} \in R^n, k \geq 0. \tag{33}$$

Furthermore, (33) with $\boldsymbol{z}^k$ substituting for $\boldsymbol{z}$ leads to

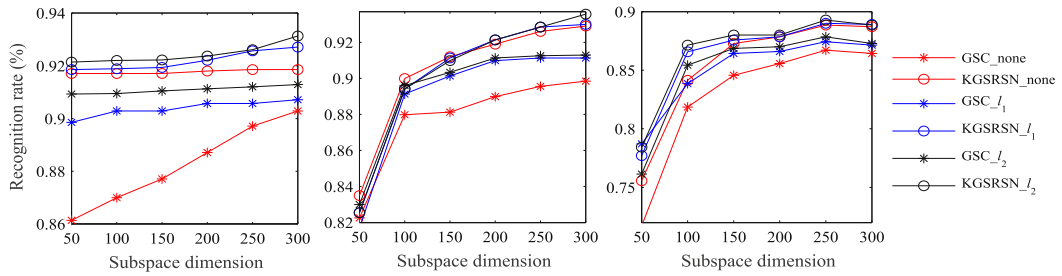
$$\frac{\delta}{2}||\boldsymbol{z}^k - \boldsymbol{z}^{k+1}||_2^2 \leq Q(\boldsymbol{z}^k, \boldsymbol{z}^k) - Q(\boldsymbol{z}^{k+1}, \boldsymbol{z}^k) \leq J(\boldsymbol{z}^k) - J(\boldsymbol{z}^{k+1}) \tag{34}$$

Summing the inequalities (34) over k, we then obtain

$$\sum_{k=0}^{\infty}||\boldsymbol{z}^k - \boldsymbol{z}^{k+1}||_2^2 \leq \frac{2}{\delta}\sum_{k=0}^{\infty}(J(\boldsymbol{z}^k) - J(\boldsymbol{z}^{k+1})) = \frac{2}{\delta}(J(\boldsymbol{z}^0) - J^*) \tag{35}$$

where J^* is the limit of the convergent sequence $\{J(\boldsymbol{z}^k)\}_{k=0}^{\infty}$ (the first statement of Proposition 5). Since that $\{J(\boldsymbol{z}^k)\}_{k=0}^{\infty}$ is a monotonically non-increasing sequence and $\delta > 0$, the last term of (35) is a finite non-negative number, which proves the convergence of the first term and verifies the property $\lim_{k\to\infty}||\boldsymbol{z}^k - \boldsymbol{z}^{k+1}||_2^2 = 0$. □

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035



(a) MNIST (b) AR (c) PIE

Fig. 3. The recognition rate of GSC and KGSRSN under different norm normalization.

For the computational complexity of Algorithm 1, it is clear to see that the main running time lies in updating $\boldsymbol{\beta}$ and $\boldsymbol{z}$. Since $(\tilde{K}+\rho\boldsymbol{I})^{-1}$ can be computed in advance and cached offline, the complexity of performing (18) is $O(n^2)$. Besides, with the fact that eacch $\boldsymbol{z}_i, i=1,\ldots,c$, can be updated in parallel as in (26), then the complexity of $\boldsymbol{z}$ subproblem costs $O(i_i c n_i)$. Therefore, the overall time complexity of Algorithm 1 is $O(i_o(n^2+i_i c n_i^2))$. In our experiments, KGSRSN always reaches the convergence condition (we set $\varepsilon_1=\varepsilon_2=1e-3$) with $i_i \le 10$ and $i_o \le 30$.

4. Experimental analysis

4.1. Datasets and experimental settings

(1) *Datasets*. Three real-world benchmark datasets, i.e., MNIST handwritten digits database, AR and PIE face databases, are used for evaluation. For AR database, we use a subset [17] including 100 individuals for a total of 700 training and 700 testing images of 60×43 pixels. The PIE database contains 41368 face images collected from 68 subjects. In accordance with AR database, we select 700 training samples and 700 test samples at random in our experiments. For MNIST database, we randomly select 50 training images for each of the 10 digits from the training set and 70 from the test set. All images in PIE and MNIST are manually cropped to 28×28 pixels.

(2) *Form of input data*. Our experiments are conducted on original gray-valued pixels and subspace projections. For the original pixels, each image is concatenated by its columns, and then all samples are arranged in a tandem array. For subspace projection samples, dimension reduction methods, including principal component analysis (PCA) and iterative nearest neighbors linear projections (INNLP) [10], are used for feature extraction. We set the regularization parameter of INNLP to be 0.05 permanently.

(3) *Competing methods*. The selected competing classifiers include linear algorithms SRC [7], CRC [8], and GSC [12], weighted algorithms WSRC [16], WCRC [17], and WGSC [21], kernel-based algorithms KSRC [24], KCRC [17], KGSC [27], and KWCRC [17]. In addition, the performance of KGSRSN is also compared with other classical algorithms, including INNC [10], NMR [22] and KINNC [10]. In the implementation, we adopt 5-fold cross validation to confirm optimal value of the regularization parameters. For arbitrary classification algorithms, we search λ from $\{1e-6, 1e-5, \ldots, 1e0\}$, and select the highest average recognition rate as the ultimate model parameters in different feature dimension. Without special declaration, in our method we set ρ=1 and a=$0.9\rho/(\lambda\ \max(r_i))$.

4.2. The behavior of normalization and non-convex penalty

As described in Section 3.1, different normalization of input data affects the performance of SR-based approaches. In this section, we take GSC and KGSRSN as examples to evaluate the recognition rate versus different norm normalization on MNIST, AR and PIE databases in Fig. 3. PCA is used for feature extraction. Fig. 3 shows the recognition rate of GSC varies greatly with the change of norm normalization, where its gap reaches 2–6% under different subspace dimensions. Among them, GSC-l_2 reaches the highest recognition rate with l_2-norm while GSC_none performs poorest. Besides, KGSRSN takes advantage of kernel trick to overcome the normalization problem described in Section 3.1, so its recognition rate changes less (about 1%) under different datasets and different subspace dimensions. Overall, the experimental results in Fig. 3. agree well with the theoretical analysis in Section 3.1. In the following experiments, we adopt l_2-norm normalization for better recognition rate of linear algorithms.

In Algorithm 1, parameter a is used to determine the degree of non-convexity for the logarithmic penalty. With Proposition 3, our experiments adopt $a=\tau\rho/(\lambda\max(r_i))$, $\tau\in\{0.1, 0.2, \ldots, 0.9\}$ to ensure the overall convexity of the total cost function. Fig. 4 illustrates the role of parameter a. In Fig. 4 (a), a specific parameter a closer to 0 makes the penalty function closer to the classical l_1-norm constraint. On the other hand, a larger τ leads to a deeper degree of non-convexity. Fig. 4 (b) illustrates the recognition rate of our method versus various choice of τ in MNIST, AR, and PIE, respectively. We can see that the performance of KGSRSN improves with the increasing of τ to some extent. This empirically verifies the statement that more non-convex penalty leads to a more compact representation.

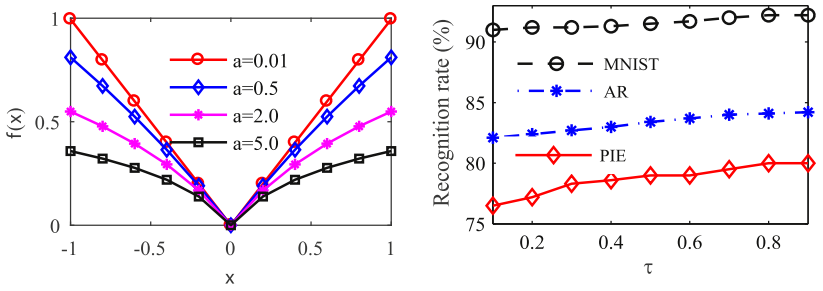
4.3. Recognition performance

In this section, we conduct classification on original input data and subspace data, respectively. The subspace dimension is set as l={50, 100, 150, 200, 250, 300}. Tables 1–3 show the recognition rates and corresponding dimension on MNIST, AR, PIE, where the best results are highlighted in bold and the second best results are highlighted in italics. From Tables 1–3, we can see that

(1) Considering diverse distribution of practical images, most classifiers have different performance in different input feature. We cannot guarantee that any classifier has overwhelming superiority, so we attempt to seek out the one that is relatively more stable and more effective.

(2) The recognition rate of INNC and KINNC is clearly lower than other methods. Take PIE database with PCA as example, they achieve 61.0% recognition rate while the lowest recognition rate of other algorithms is 64.7%. It shows that nearest neighbor based algorithms are suboptimal for classifica-

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035



(a) Logarithmic function (b) Recognition results

Fig. 4. Plot of the penalty function defined in (8) and the recognition results with various parameter values.

Table 1
Recognition rates versus different dimensions on MNIST databases.

Classifier	Input	Subspace dimensions from PCA						Subspace dimensions from INNLP					
	dimension	50	100	150	200	250	300	50	100	150	200	250	300
SRC	91.1	91.0	91.0	91.0	91.0	91.3	91.1	65.7	65.7	65.1	65.3	65.7	65.7
CRC	86.4	87.4	87.1	87.1	87.0	87.0	87.0	63.1	63.3	63.1	62.9	62.9	62.9
GSC	91.6	90.9	91.0	91.1	91.1	91.2	91.3	65.3	66.1	66.0	66.4	66.3	66.3
WSRC	89.7	90.9	90.6	90.4	90.7	91.1	90.9	65.7	64.9	63.6	65.3	65.0	65.1
WCRC	88.0	90.3	90.4	89.9	89.7	89.4	89.4	64.4	63.9	64.3	63.3	63.0	63.0
WGSC	91.6	91.5	91.6	91.6	91.6	91.6	91.6	65.3	66.8	66.8	67.0	67.0	67.3
INNC	91.6	91.6	91.9	91.7	91.7	91.6	91.6	65.6	64.1	66.9	65.0	65.3	65.3
NMR	85.8	87.2	87.2	87.2	87.1	87.1	87.1	63.2	62.2	63.2	63.0	63.0	63.0
KSRC	91.7	91.9	92.3	91.9	91.9	92.0	92.0	66.3	66.6	66.4	67.9	67.4	67.4
KCRC	91.7	91.7	90.6	90.0	89.3	88.3	88.5	**66.7**	67.9	67.1	67.9	67.3	67.4
KGSC	91.8	91.7	92.1	92.1	92.0	91.8	91.8	66.6	67.8	67.6	67.6	67.6	67.4
KWCRC	91.7	90.1	90.1	90.1	90.3	90.1	90.1	**66.7**	67.9	67.1	67.9	67.3	67.4
KINNC	91.5	91.4	91.4	91.6	91.4	91.6	91.6	55.4	48.1	42.3	40.1	36.7	36.1
KGSRSN	**92.2**	**92.2**	**92.4**	**92.8**	**92.6**	**92.6**	**92.4**	66.6	**68.4**	**68.3**	**68.3**	**67.9**	**67.9**

Table 2
Recognition rates versus different dimensions on AR database.

Classifier	Input	Subspace dimensions from PCA						Subspace dimensions from INNLP					
	dimension	50	100	150	200	250	300	50	100	150	200	250	300
SRC	93.6	76.4	79.4	80.5	81.4	81.6	81.8	90.6	93.3	93.9	94.3	94.4	94.1
CRC	93.0	77.5	87.6	89.6	90.7	91.0	92.6	91.0	94.0	93.9	93.4	93.4	93.1
GSC	94.0	83.0	88.9	90.3	91.1	91.2	91.3	92.1	**94.3**	94.4	94.3	94.1	94.3
WSRC	92.3	81.7	88.8	90.4	91.7	92.0	92.0	89.6	93.3	**95.3**	94.6	94.0	93.8
WCRC	93.0	80.8	88.1	89.7	91.3	91.7	92.3	**93.4**	94.1	94.3	94.4	94.4	94.4
WGSC	94.0	84.0	88.9	90.4	91.3	92.1	92.4	92.3	**94.3**	94.4	94.4	94.2	94.3
INNC	80.1	73.3	77.3	78.0	78.8	79.0	79.5	88.8	92.6	92.9	93.0	92.9	93.0
NMR	93.2	78.0	88.0	89.0	90.8	91.0	92.4	91.2	93.8	93.6	94.5	93.4	93.2
KSRC	81.9	75.5	78.3	79.4	80.5	80.7	81.3	87.8	92.1	92.3	92.4	92.4	92.4
KCRC	93.3	82.4	88.8	90.4	91.7	91.7	91.8	92.7	93.6	93.9	93.7	93.7	93.6
KGSC	93.8	83.0	88.9	90.6	91.7	91.8	91.8	92.6	94.0	94.0	94.2	94.6	94.4
KWCRC	92.4	78.8	86.0	89.3	91.3	91.0	92.1	85.8	92.4	92.7	92.6	92.7	92.7
KINNC	80.6	74.2	77.4	78.0	79.2	79.9	80.1	83.0	84.9	76.5	75.4	73.1	62.8
KGSRSN	**94.2**	**84.2**	**90.2**	**91.3**	**92.3**	**92.9**	**93.6**	93.2	93.9	94.7	**94.8**	**94.8**	**94.8**

Table 3
Recognition rates versus different dimensions on PIE database.

Classifier	Input	Subspace dimensions from PCA						Subspace dimensions from INNLP					
	dimension	50	100	150	200	250	300	50	100	150	200	250	300
SRC	89.0	70.7	77.4	78.4	79.6	79.6	80.6	88.6	89.7	90.7	90.4	90.9	91.1
CRC	88.0	64.7	80.1	84.9	85.9	86.4	87.6	88.4	89.4	88.7	89.0	89.1	89.0
GSC	89.0	76.1	**85.4**	**86.9**	87.0	**87.9**	87.3	88.7	89.4	89.6	89.6	89.7	89.9
WSRC	84.7	75.0	83.4	84.7	85.7	86.9	86.4	89.4	89.1	89.6	89.7	89.4	89.6
WCRC	87.9	66.6	79.3	83.6	85.0	85.4	85.3	89.7	89.6	89.7	89.9	90.0	90.1
WGSC	88.5	77.2	85.0	85.4	86.8	87.0	86.9	89.8	89.6	90.0	90.0	90.0	90.2
INNC	61.3	52.9	57.6	59.3	60.6	60.6	61.0	89.9	90.6	91.6	91.3	90.9	91.1
NMR	87.8	65.0	79.6	85.0	85.6	86.4	87.0	88.1	88.9	88.8	89.1	89.1	89.1
KSRC	80.6	72.3	74.1	75.7	76.1	76.4	77.1	88.4	88.9	88.7	88.7	88.1	88.6
KCRC	88.7	79.7	83.9	85.9	87.1	87.7	86.6	88.6	88.7	89.0	89.1	88.4	88.6
KGSC	88.9	76.5	**85.4**	**86.9**	87.2	**87.9**	87.3	89.0	89.4	89.8	88.7	88.1	88.6
KWCRC	89.0	72.6	81.6	84.0	87.3	87.6	87.0	88.6	88.7	89.0	90.0	90.0	90.0
KINNC	43.0	33.6	34.1	33.1	33.1	34.3	35.7	83.3	79.6	82.6	83.0	80.7	79.0
KGSRSN	**89.9**	**80.0**	**85.1**	86.9	**87.9**	**87.9**	**87.7**	**90.8**	**90.7**	**91.9**	**91.7**	**91.6**	**91.7**

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

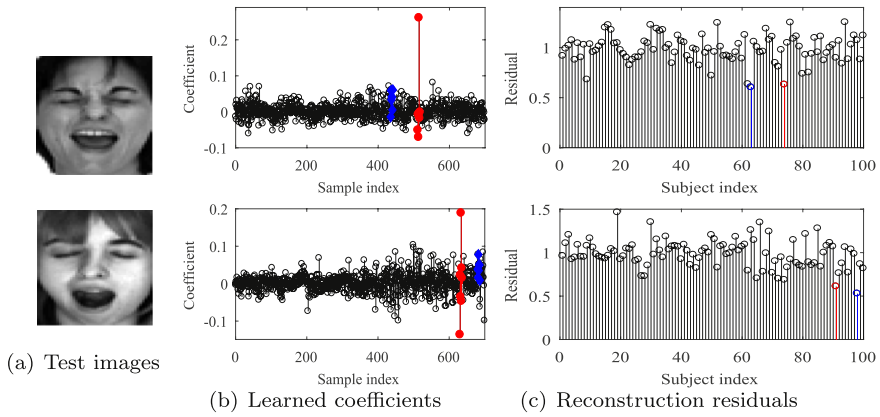


Fig. 5. The leaned coefficients and reconstruction residuals of two failed samples, where the entries from the true subject and the mistakenly identified subject are marked in red and blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tion in real world applications compared with the SR-based algorithms.

(3) Among the linear approaches, GSC outperforms SRC and CRC in most scenarios. Specifically, GSC wins 3 bold values in PIE database, which ties with KGSC in the second place. This is attributed to the group constraint with supervised $l_{2,1}$-norm. Interestingly, the performance of the matrix-based method, NMR that is developed for block corruptions, is equally matched with CRC, since that its advantages cannot be taken in these Gaussian or Laplacian distributed data [23].

(4) The performance of linear representation methods can be further improved by locality metrics and kernelization. From Tables 1–3, it can be seen that the recognition rates of weighted methods and kernel methods are mostly superior. Particularly, when the dimensionality of the data is reduced to 150 by INNLP in AR dataset, WSRC achieves 95.3% recognition rate, which outperforms all other competing algorithms.

(5) By integrating all the merits of group constraint, locality metrics, nonlinear mapping, and non-convex penalty, KGSRSN obtains the highest recognition rates in most scenarios. It achieves 92.8%, 94.8%, 91.9% recognition rate in MNIST, AR, PIE, respectively. Notice that it may not lead to better performance in practical applications with parts of these properties. In AR database, many results from the kernel methods are poorer than those from the linear methods. KGSRSN achieves the appealing performance benefiting from the properties of group constraint and locality metrics in this situation. Similarly, the performance of the weighted methods in PIE database exhibit little improvements. However, our method still performs better with the properties of group constraint and nonlinear mapping.

To further improve the performance of our method, Fig. 5 exhibits two failed tests in AR database. From Fig. 5 (a), we can see that both of these two samples are with exaggerated facial expressions, which makes them difficult to be accurately represented by their intra-class samples. In Fig. 5 (b), it is clear that if we classify the query samples to the class with the largest coefficients, then both of these two tests will be correctly identified. In Fig. 5 (c), the reconstructed residuals from the intra-class samples are very close to the minimum one. Thus, the failure also can be avoided by a majority vote from several SR-based methods. However, these tricks only work for some special samples, and cannot make an overall improvement. By introducing a learned convolutional neural network [41] as a deep features extractor, we further improve the recognition rate of KGSRSN to 98.5% and 98.3% in AR and PIE, respectively. These results are close to or even better than the deep learning based methods such as DeepFace [42]. Since our method has more intuitive learning mechanisms and can be independently applied with limited sample size, it has wide application prospects.

Table 4
Comparison of running time for recognizing one query sample.

Methods	elapsed time(in seconds)		
	MNIST	AR	PIE
SRC	0.163	0.346	0.294
GSC	0.054	0.176	0.130
WGSC	0.062	0.180	0.140
KSRC	0.182	0.255	0.138
KGSC	0.112	0.210	0.142
KWCRC	0.027	0.047	0.051
KGSRSN	0.053	0.064	0.107

4.4. Computation efficiency

In this section, several experiments are conducted to verify the efficiency of KGSRSN in comparison with six algorithms, i.e., SRC, GSC, WGSC, KSRC, KGSC and KWCRC. The programming platform is with Intel Core i5 CPU, 2.4 GHz dual-core processor, 4 GB RAM memory, 32 bits Win 7 operating system and MATLAB 2014. The average elapse time from 10 runs of recognizing one original input data for each algorithm is illustrated in Table 4. We can observe that KWCRC is the most efficient one among all the competing methods due to the closed-form solution. KSRC, which achieves similar accuracy as KWCRC, consumes about 3–5 times more time than KWCRC for recognizing one query sample. The group constrained methods, including GSC, WGSC, KGSC, and our KGSRSN, also need iterative computations as SRC in implementation, so their computational efficiency is relatively lower than KWCRC. Our KGSRSN runs faster than the other group constrained methods. This is attributed to the newly proposed optimization algorithm, which not only consumes little costs in each update step, but also converges with few loops. Fig. 6 illustrates the convergence of KGSRSN using a randomly selected sample from MNIST database. For the outer loop i_o, the objective values of our cost function (13) decrease to below 1e-3 (in log domain) within twenty iterations, and for the inner loop i_i, the convergence condition can be satisfied in around 5–10 iterations.

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035

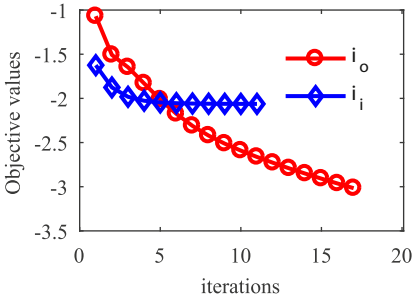


Fig. 6. The decreasing objective values of KGSRSN versus iterations i_o and i_i on MINST database.

5. Conclusion

In this paper, a new kernel group sparse representation approach via structural and non-convex constraints (KGSRSN) is proposed for classification. Specifically, three appealing properties, i.e., data locality for holding more structural information, group constraint for penalizing inter-class representation, as well as kernelization for implicitly avoiding norm normalization problem, are incorporated into a unified cost function for better discrimination. Furthermore, we introduce a non-convex function with parametric forms to penalize the representation coefficients; and we ensure an interval for the parameter that leads to the convexity of the total cost function. Experiments are conducted on benchmark databases and the results verify KGSRSN outperforms many SR-based methods. Moreover, an iteratively update solution of the convex problem for KGSRSN is also presented, which can achieve the unique solution of the algorithm within 30 iterations. Experimental results also show that the efficiency of KGSRSN is superior to that of GSC and SRC.

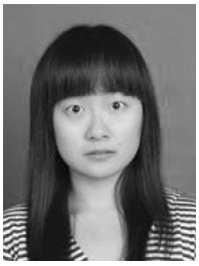
Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work was supported in part by National Natural Science Foundation of China (61602413, 61573316, 61603339) and Royal Society-Newton Mobility grant (IE151018).

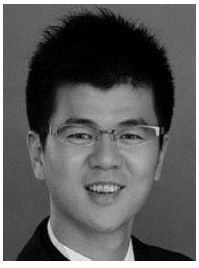
References

[1] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[2] S.W. Ma, X. Zhang, S. Wang, J. Zhang, H. Sun, W. Gao, Entropy of primitive: from sparse representation to visual information evaluation, IEEE Trans. Circuits Syst. Video Technol. 27 (2) (2017) 249–260.
[3] W.K. Yang, Z.Y. Wang, C.Y. Sun, A collaborative representation based projections method for feature extraction, Pattern Recognit. 48 (1) (2015) 20–27.
[4] W.K. Yang, C.Y. Sun, W.M. Zheng, A regularized least square based discriminative projections for feature extraction, Neurocomputing 175 (1) (2016) 198–205.
[5] S.P. Zhang, H.X. Yao, H.Y. Zhou, X. Sun, S.H. Liu, Robust visual tracking based on online learning sparse representation, Neurocomputing 100 (1) (2013) 31–40.
[6] S.P. Zhang, X.Y. Lan, H.X. Yao, H.Y. Zhou, D.C. Tao, X.L. Li, A biologically inspired appearance model for robust visual tracking, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2357–2370.
[7] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[8] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition, in: Proceedings of the 2011 International Conference on Computer Vision (2011) 471–478.
[9] J. Lai, X.D. Jiang, Classwise sparse and collaborative patch representation for face recognition, IEEE Trans. Image Process. 25 (7) (2016) 3261–3272.
[10] R. Timofte, L.V. Gool, Iterative nearest neighbors, Pattern Recognit. 48 (1) (2015) 60–72.
[11] E. Elhamifar, R. Vidal, Robust classification using structured sparse representation, Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (2011) 1873–1879.
[12] J. Huang, F.P. Nie, H. Huang, C. Ding, Supervised and projected sparse coding for image classification, Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (2013) 438–444.
[13] S.B. Tan, X. Sun, W.T. Chan, L. Qu, L. Shao, Robust face recognition with kernelized locality-sensitive group sparsity representation, IEEE Trans. Image Process. 26 (10) (2017) 4661–4668.
[14] J.W. Zheng, D. Yang, S.Y. Chen, W.W. Liang, Incremental min–max projection analysis for classification, Neurocomputing 123 (2014) 121–130.
[15] Z.Z. Fan, M. Ni, Q. Zhu, E. Liu, Weighted sparse representation for face recognition, Neurocomputing 151 (2015) 304–309.
[16] C.Y. Lu, H. Min, J. Gui, L. Zhu, Y.K. Lei, Face recognition via weighted sparse representation, J. Vis. Commun. Image Represent. 24 (2) (2013) 111–116.
[17] R. Timofte, L.V. Gool, Adaptive and weighted collaborative representation for image classification, Pattern Recognit. Lett. 43 (1) (2014) 127–135.
[18] J. Wu, R. Timofte, L.V. Gool, Learned collaborative representations for image classification, Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (2015) 456–463.
[19] Y.W. Chao, Y.R. Yeh, Y.W. Chen, Y.J. Lee, Y.F. Wang, Locality-constrained group sparse representation for robust face recognition, Proceedings of ICIP (2011). 761C764.
[20] Y. Sun, Q. Liu, J. Tang, D. Tao, Learning discriminative dictionary for group sparse representation, IEEE Trans. Image Process. 23 (9) (2014) 3816–3828.
[21] X. Tang, G. Feng, J. Cai, Weighted group sparse representation for under sampled face recognition, Neurocomputing 145 (18) (2014) 402–415.
[22] J. Yang, L. Luo, J.J. Qian, Y. Tai, F.L. Zhang, Y. Xu, Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes, IEEE Trans. Pattern Anal. Mach. Intell. 39 (1) (2017) 156–171.
[23] J.W. Zheng, P. Yang, S.Y. Chen, G.J. Shen, W.W. Liang, Iterative re-constrained group sparse face recognition with adaptive weights learning, IEEE Trans. Image Process. 26 (5) (2017) 2408–2423.
[24] L. Zhang, W.D. Zhou, P.C. Chang, J. Liu, Z. Yan, T. Wang, F.Z. Li, Kernel sparse representation-based classifier, IEEE Trans. Signal Process. 60 (4) (2012) 1684–1695.
[25] B. Wang, J. Guo, Y. Zhang, C. Li, Hierarchical feature concatenation-based kernel sparse representations for image categorization, Vis. Comput. 33 (5) (2017) 647–663.
[26] W.Y. Liu, Z.D. Yu, L.J. Lu, Y.D. Wen, H. Li, Y.X. Zou, KCRC-LCD: discriminative kernel collaborative representation with locality constrained dictionary for visual categorization, Pattern Recognit. 48 (10) (2015) 3076–3092.
[27] G. Goswami, R. Singh, M. Vatsa, A. Majumdar, Kernel group sparse representation based classifier for multimodal biometrics, Proceedings of IJCNN, IEEE, (2017) 1–8.
[28] Y. Zhang, W.Z. Ye, J.J. Zhang, Sparse signal recovery by accelerated $l_q(0<q<1)$ thresholding algorithm, Int. J. Comput. Math. 10.1080/00207160.2017.1284314
[29] J.F. Determe, J. Louveaux, L. Jacques, F. Horlin, On the noise robustness of simultaneous orthogonal matching pursuit, IEEE Trans. Signal Process. 65 (4) (2017) 864–875.
[30] S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, SIAM J. Numer. Anal. 49 (6) (2011) 2543–2563.
[31] H. Yan, J. Yang, Sparse discriminative feature selection, Pattern Recognit. 48 (5) (2015) 1220–1227.
[32] Z. Zheng, X. Huang, Z. Chen, X. He, H. Liu, J. Yang, Regression analysis of locality preserving projections via sparse penalty, Inf. Sci. 303 (2015) 1–14.
[33] C.Y. Lu, J.H. Tang, S.C. Yan, Z.C. Lin, Nonconvex nonsmooth low-rank minimization via iteratively reweighted nuclear norm, IEEE Trans. Signal Process. 25 (2) (2016) 829–839.
[34] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[35] J. Singh, D. Kumar, R. Swroop, Numerical solution of time- and space-fractional coupled burgers' equations via homotopy algorithm, Alex. Eng. J. 55 (2) (2016) 1753–1763.
[36] K. Zhang, L. Zhang, M.H. Yang, Fast compressive tracking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 2002–2015.
[37] C. He, C. Hu, X. Li, X. Yang, W. Zhang, A parallel alternating direction method with application to compound l1-regularized imaging inverse problems, Inf. Sci. 348 (2016) 179–197.
[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011). 1C112
[39] A. Lanza, S. Morigi, I. Selesnick, F. Sgallari, Nonconvex nonsmooth optimization via convex–non convex majorization minimization, Numer. Math. 136 (2) (2017) 343–381.
[40] J. Mairal, Incremental majorization–minimization optimization with application to large-scale machine learning, SIAM J. Optim. 25 (2) (2015) 829–855.
[41] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, Proceedings of British Machine Vision Conference(2015). 41.1–41.12.
[42] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deep-face: closing the gap to human-level performance in face verification, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2014) 1701–1708.

Jianwei Zheng received M.S. degree in Electrical and Information Engineering in 2005 and Ph.D. degree in Control Theory and Control Engineering in 2010 from Zhejiang University of Technology, China. Since 2010, he has been with the college of computer science, Zhejiang University of Technology. Currently he is a research fellow with the National Centre for Computer Animation, Bournemouth University, U.K. His research interest covers sparse coding, low-rank decomposition, and non-convex optimization.

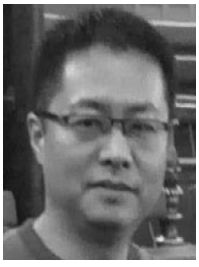


Hong Qiu was born in Ningbo, China and went to the Zhejiang University of Technology, where she studied computer science and technology and obtained her undergraduate degree in 2012. Now she is a doctoral student at Zhejiang University of Technology. Her research interest covers image processing and machine learning. She has a practical and realistic attitude towards science and a creative spirit.



Weiguo Sheng received the M.Sc. degree in information technology from the University of Nottingham, U.K., in 2002 and the Ph.D. degree in computer science from Brunel University, U.K., in 2005. Then, he worked as a Researcher at the University of Kent, U.U. and Royal Holloway, University of London, U.K. He is currently a Professor at Hangzhou Normal University. His research interests include evolutionary computation, data mining/clustering, pattern recognition and machine learning.

Xi Yang received the B.E. degree from Xian Jiaotong University, China, in 2004; the M.E. degree from Zhejiang University, China, in 2007; and the Ph.D. degree from the Chinese University of Hong Kong, HKSAR, China, in 2011. Since 2012, he has been with the college of computer science and engineering, Zhejiang University of Technology. His research interests include nonlinear control and optimization with applications in nonlinear output regulation, multi-agent systems, intelligent transportation systems, and image processing.



Hongchuan Yu (Ph.D. (2000), M.Sc. (1996), B.Sc. (1990)) is a Senior Lecturer of computer graphics in National Centre for Computer Animation, Bournemouth University. He received his Ph.D. in Computer Vision, Inst. of Intelligent Machine, Chinese Academy of Sciences, in 2000. After that, he worked as research fellow at Tsinghua University; Nanyang Technological University (Singapore) and The University of Western Australia (Perth). As principal investigator, he has secured over £2 million in research grants from EU FP7, EU H2020, EPSRC etc. He has published more than 70 academic articles in reputable journals and conferences, and regularly served as PC members/referees for international journals and conferences, including IEEE, TPAMI, IEEE, TIP, IEEE, TVCG, IVC, PR, CVIU, PRL, CAD, CGI, etc.

Please cite this article as: J. Zheng et al., Kernel group sparse representation classifier via structural and non-convex constraints, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.03.035