Sparse Group Regression Classification for Face Recognition

Chengyong Zheng* and Zhiyu Luo
School of Mathematics and Computational Science, Wuyi University, Jiangmen, China

*Corresponding author email: cyZheng@wyu.edu.cn

Abstract. Face recognition has been widely used in biometric verification which is a significant issue in system control in computer based communication. Sparse regression classification (SRC) has been an important method for face recognition. SRC assumes its regression coefficients are globally sparse. However, these regression coefficients correspond to a class-specific gallery may be locally dense. In order to get some regression coefficients that are globally sparse and locally dense, sparse group regression (SGR) is introduced in this paper for face recognition. By use of SGR, we expect to get some regression coefficients which are not only sparse as a whole, but also locally dense in some part. Variable splitting and alternating direction method of multipliers (ADMM) are used for solving the problem of SGR. The proposed algorithm, named Sparse Group Regression Classification (SGRC), is extensively evaluated on four standard databases. A comparative study with some state-of-the-art algorithms reflects the efficacy of the proposed approach.

1. Introduction
Face recognition (FR) has recently received significant attention, especially during the past several years. Various face recognition technology has been proposed within the last two decades, among of which, regression-based classification has been used frequently in FR [1][2][3][4]. In [1], Wright et al. reported a sparse representation-based classification (SRC) for robust FR. In the scheme of SRC, a probe image is first sparsely represented by all gallery images, and the classification is then achieved by finding the class that yields the least coding error. Literature [1] boosts the research of sparsity-based pattern classification [5-7]. However, some researchers have questioned the use of sparsity in image classification, see [8] for example.

By using the concept that patterns from a single-object class lie on a linear subspace and a probe image can be represented as a linear combination of class-specific galleries, literature [2] reported a linear regression classification (LRC) algorithm for FR. LRC is reported as a fairly simple but efficient algorithm for FR. With LRC and the kernel trick, a kernel linear regression classification (KLRC) is reported in [3]. Literature [4] reported a collaborative representation-based classification method using k-nearest classes. In [9], a group-sparse representation variational method is applied to FR.

SRC assumes a probe image can be sparsely represented by all gallery images, and LRC uses the concept that a probe image can be well (densely) represented by some gallery images that of the same class as the probe image. Let $y$ be the feature of a probe image, and $A$ ($i = 1,...,k$ $k$ is the total number of classes) be the matrix constituted by all the features of the gallery images that belong to the $i$-th class. Let $Ax = [A_1 A_2 \cdots A_k]$. SRC tries to find the most sparse $x$, subject to $Ax = y$ or $||Ax - y||_2 < \varepsilon$ ($\varepsilon$ is a pre-given small positive constant). However, LRC tries to find $x^*$, subject to...
$$x_i^* = \arg \min_{x_i} \| A_i x_i - y \|_2^2 \quad (i = 1, \ldots, k)$$

Usually, $x_i^*$ is not sparse. Taking the results of SRC and LRC into account, embedding $x_i^*$ to the $i$-th part of $x$ shows that a better $x$ that meets $A x = y$ or $\| A x - y \| < \epsilon$ may be globally sparse and locally dense as well. Inspired by this hint, by blending the $L_1$ norm with the $L_2$ norm, this paper presents a sparse group regression classification (SGRC) for face recognition.

Sparse group regression (SGR) not only enforces sparseness on the groups, but also on the singleton variables, so SGR can yield solutions that are sparse at both the group and individual feature levels. Because of the potential of SGR to produce regression coefficients that are globally sparse and locally dense as well, this paper introduces SGR for face recognition.

The main purpose of this paper is to illustrate the potential of the sparse group regression (SGR) technique in classification problems, and to introduce an optimization algorithm to solve efficiently the convex optimization problem underlying the SGR technique.

The rest of the paper is organized as follows: In Section 2, the proposed SGRC algorithms are described. Extensive experiments using standard databases are reported in Section 3. Section 4 concludes this paper.

2. Classification Based on Sparse Group Regression

2.1. Sparse Group Regression

For a supervised classification problem, we firstly need to choose some training samples for each class. Let $k$ be the total number of classes, $n_i$ be the number of training samples of the $i$-th ($i = 1, \ldots, k$) class, and $n = \sum_{i=1}^{k} n_i$ be the total number of training samples. Let $A_i = [a_{i1}, a_{i2}, \ldots, a_{in_i}]$, where $a_j \in \mathbb{R}^d$ is the feature vector of the $j$-th ($j = 1, \ldots, n_i$) training sample of the $i$-th class. Let $A = [A_1, A_2, \ldots, A_k]$. Then if a test sample $y \in \mathbb{R}^d$ belongs to the $i$-th class, $y$ can be represented as a linear combination of $A_i$, i.e., $y = A_i x_i$, where $x_i \in \mathbb{R}^{n_i}$ is the regression coefficient. The coefficient $x_i$ is not always sparse, it may be dense. On the whole, a suitable solution $x$ of system $A x = y$, as shown in Fig. 1, can be expected to be globally sparse and locally dense. That is, if $y$ belongs to the $i$-th class, $x$ should be sparse at most parts except the part corresponds to the $i$-th class.

![Figure 1. Group sparsity of $x$ when a test sample belongs to the $i$-th class.](image)

But we do not know which class $y$ belongs to in advance. For a test sample $y$ of unknown class, coefficient $x$ can still be expected to be globally sparse and dense at a part corresponds to a certain class. Based on this speculation, we adopt the following optimization problem

$$\min_x \lambda_1 \| x \|_1 + \lambda_2 \| x \|_{2,1}^2, s.t. A x = y,$$

(1)

where $\| x \|_{2,1}^2 = \sum_{i=1}^{k} \| x_i \|_2$, $\lambda_1$ and $\lambda_2$ are two trade-off parameters balancing the global sparsity and the local density of $x$.

In the case of Gaussian noise, problem (1) can be rewritten as follows

$$\min_x \lambda_1 \| x \|_1 + \lambda_2 \| x \|_{2,1}^2 + \frac{1}{2} \| A x - y \|_2^2.$$

(2)

Now let us consider the situation of multitasking. Let $m$ be the total number of test sample, $y_j \in \mathbb{R}^d$ ($j = 1, \ldots, m$) be the feature vector of the $j$-th test sample. Let $Y = [y_1, \ldots, y_m]$. It is not hard to see that (5) is equal to
3  

\[ \min_{X} \sum_{j=1}^{m} \left[ \lambda_1 \| X_j \|_1 + \lambda_2 \| X_j \|_{\ell_2,1}^2 + \frac{1}{2} \| AX_j - y_j \|_2^2 \right] \]  

(3)

which is also equal to solve \( m \) optimization problems:

\[ X_j = \arg \min_{X} \lambda_1 \| X_j \|_1 + \lambda_2 \| X_j \|_{\ell_2,1}^2 + \frac{1}{2} \| AX_j - y_j \|_2^2 \quad (i = 1, \ldots, m) \]  

(4)

So, in order to test all the \( m \) test samples, we only need to solve the following problem

\[ \min_{X} \lambda_1 \| X \|_{1,1} + \lambda_2 \| X \|_{\ell_2,1}^2 + \frac{1}{2} \| AX - Y \|_F^2, \]  

(5)

where \( \| X_j \|_{\ell_2,1} = \sum_{j=1}^{m} \| X_j \|_{\ell_2,1} \), \( X_j \) denotes the \( j \)-th column of \( X \), and \( \| \cdot \|_F \) denotes the Frobenius norm.

2.2. Algorithm for Sparse Group Regression

In order to solve problem (5), variable splitting and alternating direction method of multipliers (ADMM) \cite{10} are used.

First, two auxiliary variables \( Z_1 \) and \( Z_2 \) are introduced to transform problem (5) to

\[ \min_{X,Z_1,Z_2} \lambda_1 \| Z_1 \|_{1,1} + \lambda_2 \| Z_2 \|_{\ell_2,1}^2 + \frac{1}{2} \| AX - Y \|_F^2, \quad s.t. Z_1 = X, \quad Z_2 = X. \]  

(6)

Define the Lagrangian function of problem (6) as

\[ L(X, Z_1, Z_2, D_1, D_2) = \lambda_1 \| Z_1 \|_{1,1} + \lambda_2 \| Z_2 \|_{\ell_2,1}^2 + \frac{1}{2} \| AX - Y \|_F^2 + \mu \| X - Z_1 + D_1 \|_F^2 + \| X - Z_2 + D_2 \|_F^2, \]  

(7)

where \( \mu \) is a penalty parameter, \( D_1 \) and \( D_2 \) are two Lagrangian multipliers.

Based on ADMM, an iterative scheme can be described as follows:

\[ Z_1^{k+1} = \arg \min_{Z_1} L(X^k, Z_1, Z_2^k, D_1^k, D_2^k); \]  

(8)

\[ Z_2^{k+1} = \arg \min_{Z_2} L(X^k, Z_1^k, Z_2, D_1^k, D_2^k); \quad \]  

(9)

\[ X_2^{k+1} = \arg \min_{X} L(X, Z_1^{k+1}, Z_2^{k+1}, D_1^k, D_2^k); \]  

(10)

\[ D_1^{k+1} = D_1^k + (X^{k+1} - Z_1^{k+1}), \]  

(11)

\[ D_2^{k+1} = D_2^k + (X^{k+1} - Z_2^{k+1}). \]  

(12)

Formula (8) has a unique minimizer given by

\[ Z_1^{k+1} = S_{\lambda_1/\mu}(X^k + D_1^k), \]  

(13)

where \( S \) denotes the element-wise shrinkage operator defined by:

\[ S_{\tau}(\cdot) = \text{sgn}(\cdot) \max\{|\cdot| - \tau, 0\}, \quad \text{where } \tau > 0. \]  

(14)

Formula (9) can be rewritten as

\[ X_2^{k+1} = \text{arg} \min_{X} \left[ \frac{1}{2} \| AX - Y \|_F^2 + \mu \| X - Z_2^{k+1} + D_2^k \|_F^2 \right]. \]  

(15)
\[ Z_{2}^{k+1} = \arg \min_{z^{2}} \sum_{j=1}^{k} \sum_{i=1}^{m} \left( \frac{\lambda_{i}}{\mu} \| (Z_{j}^{2})_{j}^{y} \|_{2} + \frac{1}{2} \| (Z_{2}^{k} - (X^{k} + D_{2}^{k})_{j}^{y}) \|_{2}^{2} \right), \]  

where \( [\cdot]_{j}^{y} \) denotes the \( j \)-th group of the \( j \)-th column of a matrix. So

\[ [Z_{2}^{k+1}]_{j} = \arg \min_{z} \frac{\lambda_{j}}{\mu} \| z \|_{2} + \frac{1}{2} \| z - (X^{k} + D_{2}^{k})_{j}^{y} \|_{2}, i = 1,...,k, j = 1,...,m, \]  

the solution of which is (see, e.g., [11])

\[ [Z_{2}^{k+1}]_{j} = \text{vect-soft}(b, \tau)(X^{k} + D_{2}^{k})_{j}^{y} / \mu, i = 1,...,k, j = 1,...,m, \]  

where \( \text{vect-soft} \) is the vect-soft threshold function defined by

\[ \text{vect-soft}(b, \tau) = \frac{\max\{b - \tau, 0\}}{\max\{b - \tau, 0\} + \tau} \]

From formula (10), we can get

\[ X^{k+1} = (\frac{1}{\mu} A^{T} A + 2I)^{-1} (\frac{1}{\mu} A^{T} Y + Z_{1}^{k+1} + Z_{2}^{k+1} - D_{1}^{k} - D_{2}^{k}), \]  

where \( I \) denotes the identity matrix.

Summing up the above analysis, we get Algorithm 1.

**Algorithm 1 Algorithm for Sparse Group Regression**

| Input: \( X_{0}, \lambda_{1}, \lambda_{2} > 0 \), and \( \mu > 0 \) |
| Initialization: \( D_{0}^{1} = D_{0}^{2} = 0, k = 0 \) |
| repeat |
| a) get \( Z_{k+1}^{1}, Z_{k+1}^{2}, X^{k+1} \) by formula (13), (17), (18) respectively; |
| b) update \( D_{k+1}^{1} = D_{k}^{1} + (X^{k+1} - Z_{k}^{1+1}) \) and \( D_{k+1}^{2} = D_{k}^{2} + (X^{k+1} - Z_{k}^{2+1}) \); |
| c) \( k \leftarrow k + 1 \). |
| until \( \max_{i,j} |X_{i}^{k+1} - X_{j}^{k}| < \varepsilon \). |
| Output: \( X_{k} \). |

**2.3. SGRC**

Given a new test sample \( y \), its SGR representation \( \hat{x} \) is first computed by (5). By denoting \( \hat{x}_{i} \) as the coefficients associated with the \( i \)-th class, we approximate \( y \) by \( \hat{y}_{i} = A_{i} \hat{x}_{i} \). Then \( y \) can be identified by the minimum residual:

\[ \min_{i} \eta_{i}(y) = \| y - A_{i} \hat{x}_{i} \|_{2}. \]  

**Algorithm 2 SGRC**

| Input: a matrix of training samples \( A = [A_{1}, A_{2}, \cdots, A_{k}] \) for \( k \) classes, a matrix of \( p \) test samples \( Y = [y_{1}, y_{2}, \cdots, y_{p}] \). |
| 1. Solve the SGR problem: |
| \[ \hat{X} = \arg \min_{X} \lambda_{1} \| X \|_{1,1} + \lambda_{2} \| X \|_{2,2} + \frac{1}{2} \| AX - Y \|_{F}^{2}. \]  

---

4
2. Compute the residuals: \( r_i(y_j) = |y_j - A_i x_i| \) for \( i = 1, ..., k, j = 1, ..., p \).

Output: \( \text{identity}(y_j) = \arg\min_i r_i(y_j) \) for \( j = 1, ..., p \).

When \( A = 0 \), (5) or (20) becomes the problem of group Lasso [12], and Algorithm 2 (SGRC) is named as Group Lasso Classification (GLC) accordingly in this paper.

3. Experiments

In this section, extensive comparative experiments between the proposed algorithm and LRC [2], SRC [1], and GLC are carried out to verify the proposed algorithm.

3.1. Experimental Settings

We first introduce the related competing algorithms, databases, and parameter settings.

For SRC, GLC and SGRC, the regularization parameters are tuned by searching from the discrete set \( \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\} \) in each test and the best results are reported, \( \mu \) is fixed as \( 10 \lambda \). The max iteration number for SRC, GLC and SGRC are set by 200.

Four standard databases, i.e., the AT&T, FERET [13], Extended Yale B [14], and the PIE database from CMU [15] have been addressed. AT&T Database consists of a total of 400 face images, of a total of 40 people (10 samples per person). For Yale Database, we choose the frontal pose, and thus get 64 images for each person. For AT&T and Yale Database, all the face images are manually aligned and cropped, each image size of which is 32 by 32 pixels. CMU PIE Database contains 68 subjects with 41368 face images as a whole. We choose the frontal pose with varying lighting and illumination which leaves us about 49 images per subject. Each image is resized to have 32×32 pixels. FERET Database contains 1400 images of 200 human subjects under different poses, expression and illumination conditions. The size of each cropped face image is 80 by 80 pixels.

For each database, the image set is partitioned into the gallery and probe set with different percentages. For different \( p \in (0,1) \), 100p% images of each person are randomly selected for training and the rest are for testing. For each database, with every percentage, average results of 10 random splits are reported, as well as the standard deviation.

**Table 1.** Comparisons of recognition accuracy on AT&T Database (mean ± std-dev%)

| \( p \) | LRC | SRC | GLC | SGRC |
|-------|-----|-----|-----|------|
| 0.4   | 88.08 ± 1.56 | 92.58 ± 1.40 | 92.71 ± 1.56 | 92.92 ± 1.33 |
| 0.6   | 94.37 ± 2.20 | 96.31 ± 1.43 | 96.38 ± 1.56 | 96.69 ± 1.32 |
| 0.8   | 96.00 ± 2.11 | 97.62 ± 1.90 | 98.12 ± 1.56 | 98.37 ± 1.19 |

**Table 2.** Comparisons of recognition accuracy on Yale Database (mean ± std-dev%)

| \( p \) | LRC | SRC | GLC | SGRC |
|-------|-----|-----|-----|------|
| 0.4   | 93.68 ± 0.46 | 97.69 ± 0.31 | 97.37 ± 0.33 | 97.72 ± 0.34 |
| 0.6   | 95.15 ± 0.58 | 98.59 ± 0.23 | 98.43 ± 0.31 | 98.71 ± 0.23 |
| 0.8   | 96.00 ± 0.41 | 99.22 ± 0.25 | 99.24 ± 0.41 | 99.41 ± 0.35 |

where std-dev means standard deviation. As mentioned, each of the means and the standard deviations are got by averaging the results over 10 random splits.

From Table 1, we can see:

- In term of recognition accuracy, for all three cases that \( p = 0.4, 0.6 \) and 0.8, SRC outperforms LRC, but SGRC and GLC outperform LRC and SRC, and SGRC outperform all the other three methods.

- In all three cases, the std-dev values of GLC and SGRC are lower than those of LRC and SRC, which shows that the performance of GLC and SGRC are more stable than that of LRC and SRC. When \( p = 0.8 \), SGRC has the lowest std-dev value.

Table 2 and Table 3 show that:
– In terms of recognition accuracy, in all three cases when  \( p = 0.4, 0.6 \) and  \( 0.8 \), SGRC outperforms LRC, SRC and GLC, GLC and SRC outperform LRC, GLC can be hard to tell from SRC, and LRC still shows the worst recognition accuracy.

– With respect to std-dev, the values shown in Table 2 and Table 3 are significantly lower than those shown in Table 1. In term of std-dev values, no algorithm consistently outperforms the others.

**Table 3. Comparisons of recognition accuracy on PIE Database (mean ± std-dev%)**

|   | LRC   | SRC   | GLC   | SGRC   |
|---|-------|-------|-------|--------|
| 0.4| 97.48 ± 0.36 | 98.11 ± 0.25 | 98.17 ± 0.25 | 98.27 ± 0.26 |
| 0.6| 98.12 ± 0.22 | 98.36 ± 0.31 | 98.35 ± 0.24 | 98.48 ± 0.29 |
| 0.8| 98.07 ± 0.23 | 98.60 ± 0.30 | 98.34 ± 0.25 | 98.65 ± 0.22 |

**Table 4. Comparisons of recognition accuracy on FERET Database (mean ± std-dev%)**

|   | LRC   | SRC   | GLC   | SGRC   |
|---|-------|-------|-------|--------|
| 0.4| 38.14 ± 0.91 | 53.64 ± 1.52 | 98.17 ± 1.39 | 98.27 ± 1.63 |
| 0.6| 44.83 ± 1.23 | 62.35 ± 1.66 | 98.35 ± 1.55 | 98.48 ± 1.61 |
| 0.8| 53.70 ± 2.35 | 72.35 ± 2.51 | 98.34 ± 2.55 | 98.65 ± 2.30 |

In Table 4, all the mean values of recognition accuracy are lower than those shown in the aforementioned tables. But in term of recognition accuracy, LRC is still the worst one, and the proposed SGRC remains the best method. For example, when  \( p = 0.4 \) and  \( p = 0.6 \), the mean values of recognition accuracy of SGRC are more than one percent higher than those of GLC and SRC, and up more than ten points higher than those of LRC.

From the experimental results shown above, the following conclusion can be drawn:

– On four standard databases, with three different percentages of training samples, in term of recognition accuracy, the proposed SGRC consistently outperforms LRC, SRC and GLC, LRC has the worst recognition accuracy, and GLC can be hard to tell from SRC.

– On AT&T Database, Yale Database and PIE Database, all the mean values of recognition accuracy of four algorithms are much high, there is little space for further improvement. So, though the mean values of recognition accuracy of SGRC are slightly higher than those of GLC and SRC, we think they still make sense.

4. Conclusion

In this paper, a sparse group regression (SGR) based classification are proposed. SGR can yield solutions that are sparse at both group and individual feature levels and can overcome the shortcoming of SRC that can only enforce sparseness on singleton variables. Variable splitting and ADMM are used to solve SGR. Competitive face recognition experiments have carried out to show that, in term of recognition accuracy, the proposed algorithm consistently outperforms LRC, SRC and GLC.

References

[1] J. Wright, A.Y. Yang, A. Ganesh, etc. Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Machine Intell., 31(2):210-227, 2009.

[2] I. Naseem, R. Togneri, M. Bennamoun. Linear regression for face recognition. IEEE Trans. Pattern Anal. Machine Intell., 32(11):2106-2112, 2010.

[3] Yuwu Lu, Xiaozhao Fang, Binglei Xie. Kernel linear regression for face recognition. Neural Computing and Applications, 24(7-8):1843-1849, 2013.

[4] Cheng Yong Zheng, Ningning Wang. Collaborative representation with k-nearest classes for classification. Pattern Recognition Letters, 117: 30-36, 2019.

[5] Hao Hongxing, Wu Lingda. A new complex valued dictionary learning method for group-sparse representation. Optik,196: 163150, 2019.

[6] Cencheng Shen, Li Chen, Yuexiao Dong, etc. Sparse Representation Classification Beyond ℓ1 Minimization and the Subspace Assumption. IEEE Trans. Inform. Theory, 66(8): 5061-5071, 2020.
[7] Ruijing Li, Lan Tang, Yechao Bai, etc. Group-Based Sparse Representation Based on $l_p$-Norm Minimization for Image Inpainting. *IEEE Access*, 8: 60515-60525, 2020.

[8] Qinfeng Shi, A. Eriksson, A. van den Hengel, etc. Is face recognition really a Compressive Sensing problem? In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 553-560, 2011.

[9] Fritz Keinert, Damiana Lazzaro, Serena Morigi. A Robust Group-Sparse Representation Variational Method with Applications to Face Recognition. *IEEE Trans. on Image Process*, 28(6): 2785-2798, 2019.

[10] Stephen Boyd, Neal Parikh, Eric Chu, etc. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1-122, 2011.

[11] Cheng Yong Zheng, Hong Li, Qiong Wang, etc. Reweighted sparse regression for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sensing*, 99:1–10, 2015.

[12] Ming Yuan, Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49-67, 2006.

[13] P Jonathon Phillips, Harry Wechsler, Jeffrey Huang, etc. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998.

[14] Kuang-Chih Lee, Jeffrey Ho, David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(5):684–698, 2005.

[15] Terence Sim, Simon Baker, Maan Bs. The CMU pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(12):1615–1618, 2003.