Lecture 4
Sparse Representation based Classification

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Fall 2023

Sparse representation based approach

- Motivations
  - Signals are sparse in some selected domain
  - It has strong physiological support

Sparse representation based approach

- SR-based face recognition
  - It was proposed in [1]
  - In such a system, the choice of features is no longer crucial
  - It is robust to occlusion and corruption

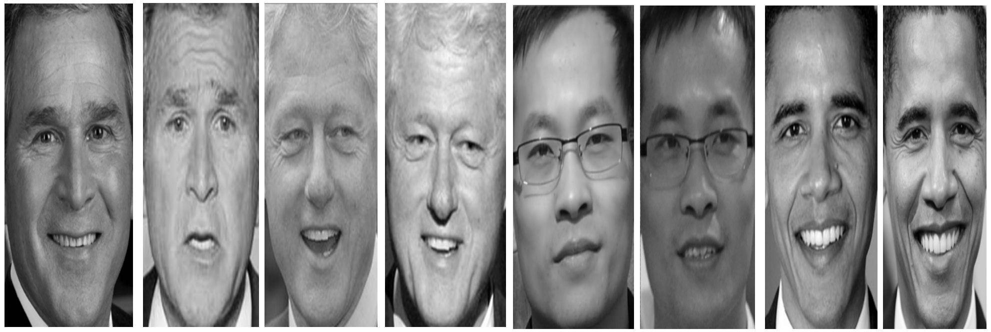
[1] J. Wright et al., Robust face recognition via sparse representation, IEEE Trans. PAMI, vol. 31, no. 2, 2009

Sparse representation based approach

- Illustration

$\mathbf{y}$

training samples



$\mathbf{v}_{1,1}$ $\mathbf{v}_{1,2}$ $\mathbf{v}_{2,1}$ $\mathbf{v}_{2,2}$ $\mathbf{v}_{3,1}$ $\mathbf{v}_{3,2}$ $\mathbf{v}_{4,1}$ $\mathbf{v}_{4,2}$

If training samples are abundant, $\mathbf{y}$ can be linearly represented by the training samples as

$$\mathbf{y} = \alpha_{1,1}\mathbf{v}_{1,1} + \alpha_{1,2}\mathbf{v}_{1,2} + \alpha_{2,1}\mathbf{v}_{2,1} + \alpha_{2,2}\mathbf{v}_{2,2}$$

$$+\alpha_{3,1}\mathbf{v}_{3,1} + \alpha_{3,2}\mathbf{v}_{3,2} + \alpha_{4,1}\mathbf{v}_{4,1} + \alpha_{4,2}\mathbf{v}_{4,2}$$

We expect that all the coefficients are zero except $\alpha_{3,1}, \alpha_{3,2}$

Sparse representation based approach

- Problem formulation

We define a matrix $\mathbf{A}$ for the n training samples of all k object classes

$$\mathbf{A} = \left[A_1, A_2, \ldots, A_k \right] = \left[\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \ldots, \mathbf{v}_{k,n_k} \right]$$

Then, the linear representation of a testing sample $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{A}\mathbf{x}_0$$

where $\mathbf{x}_0 = \left[0, \ldots, 0, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}, 0, \ldots, 0 \right]^T \in \mathbb{R}^n$

is a coefficient vector whose entries are zero except those associated with the ith class

Sparse representation based approach

This motivates us to seek the most sparsest solution to $\mathbf{y} = \mathbf{A}\mathbf{x}$, solving the following optimization problem:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \left\|\mathbf{x}\right\|_0, s.t., \left\|\mathbf{A}\mathbf{x} - \mathbf{y}\right\|_2 \leq \varepsilon \quad (1)$$

where $\left\|\cdot\right\|_0$ denotes the l_0-norm, which counts the number of non-zero entries in a vector.

However, solving (1) is a NP-hard problem, though some approximation solutions can be found efficiently.

Thus, usually, (1) can be rewritten as a l_1-norm minimization problem

Sparse representation based approach

If the solution $\mathbf{x}_0$ is sparse enough, the solution of l_0-minimization problem is equal to the solution to the following l_1-norm minimization problem:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1, s.t., \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon \quad (1)$$

The above minimization problem could be solved in polynomial time by standard linear programming methods.

There is an equivalent form for (1)

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1, \lambda > 0 \quad (2)$$

Several different methods for solving l_1-norm minimization problem in the literature, such as the l_1-magic method (refer to the course website)

Sparse representation based approach

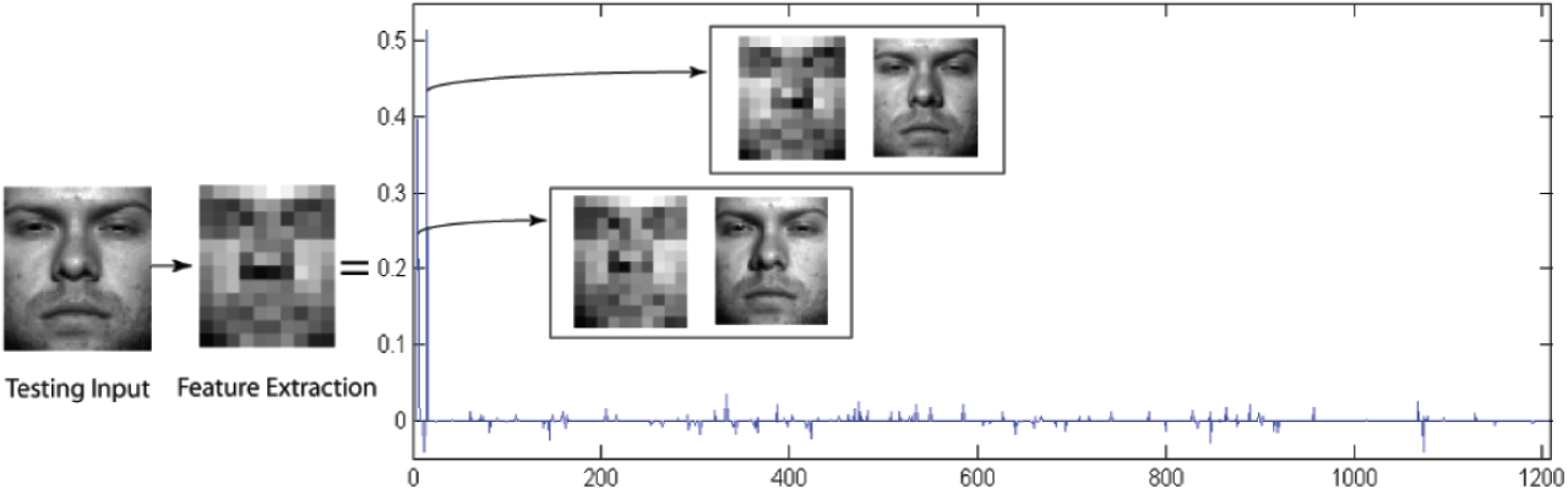
Algorithm

1. **Input**: a matrix of training samples $\mathbf{A}=\left[A_1,A_2,...,A_k\right]\in\mathbb{R}^{m\times n}$ for k classes; $\mathbf{y}\in\mathbb{R}^m$, a test sample; and an error tolerance $\varepsilon>0$
2. Normalize the columns of $\mathbf{A}$ to have unit l_2-norm
3. Solve the l_1-minimization problem
$$\mathbf{x}_0=\arg\min_{\mathbf{x}}\|\mathbf{x}\|_1, s.t., \|\mathbf{Ax}-\mathbf{y}\|_2\le\varepsilon$$
4. Compute the residuals $r_i(\mathbf{y})=\|\mathbf{y}-\mathbf{A}\delta_i(\mathbf{x}_0)\|_2$, i = {1,...,k}
5. **Output**: $\mathrm{identity}(\mathbf{y})=\mathrm{argmin}_i r_i(\mathbf{y})$

For $\mathbf{x}\in\mathbb{R}^n$, $\delta_i(\mathbf{x})\in\mathbb{R}^n$ is a new vector whose only non-zero entries are the entries in $\mathbf{x}$ that are associated with class i

Sparse representation based approach

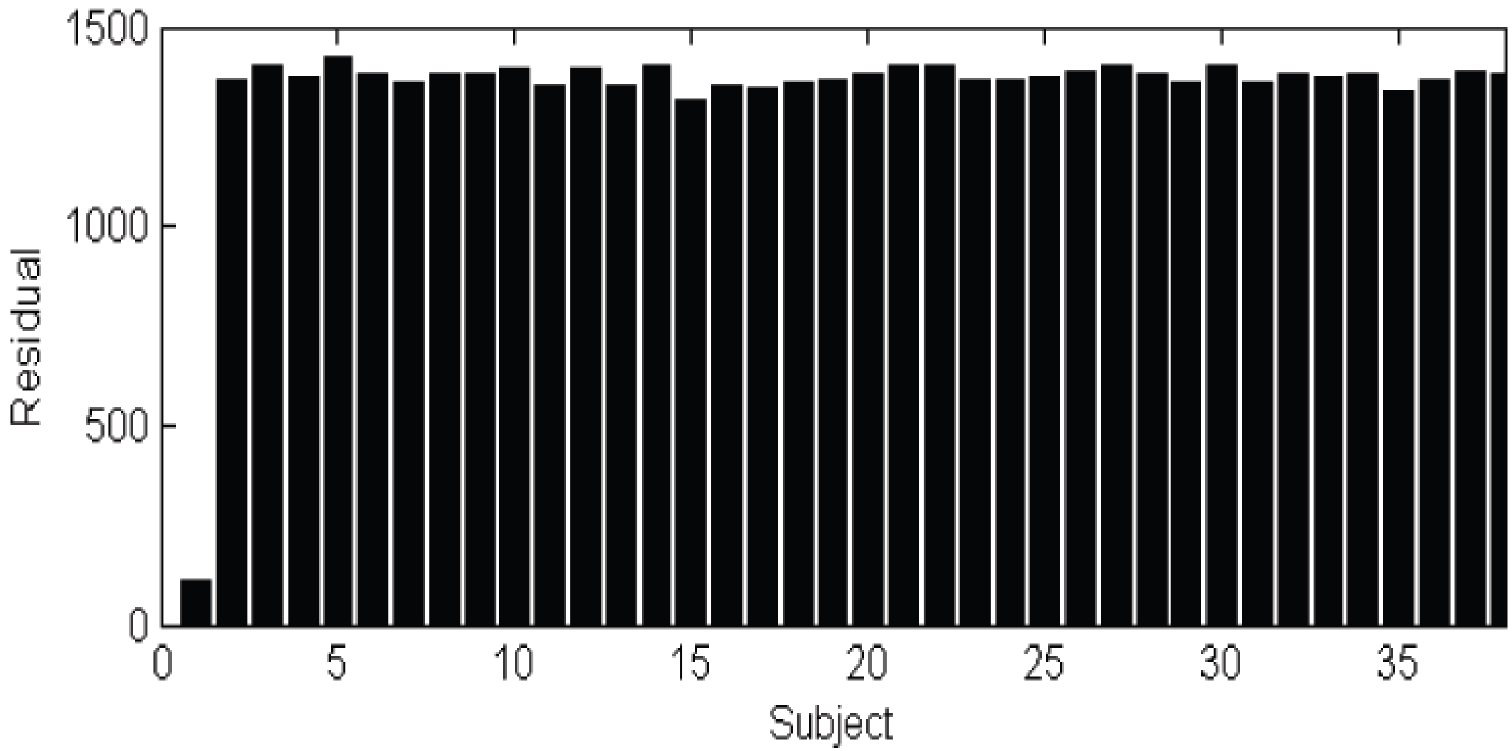
- Illustration



A valid test image. Recognition with 12 × 10 downsampled images as features. The test image $\mathbf{y}$ belongs to subject 1. The values of the sparse coefficients recovered are plotted on the right together with the two training examples that correspond to the two largest sparse coefficients.

Sparse representation based approach

- Illustration



The residuals $r_i(\mathbf{y})$ of a test image of subject 1 with respect to the projected sparse coefficients $\delta_i(\mathbf{x}_0)$ by l_1-minimization.

Sparse representation based approach

- Summary
  - It provides a novel idea for face recognition
  - By solving the sparse minimization problem, the “position” of the big coefficients can indicate the category of the examined image
  - It is robust to occlusion and partial corruption

CRC_RLS

- Collaborative representation based classification with regularized least square was proposed in [1]
- Motivation
  - SRC method is based on l_1-minimization; however, l_1-minimization is time consuming. So, is it really necessary to solve the l_1-minimization problem for face recognition?
  - Is it l_1-minimization or the collaborative representation that makes SRC work?

[1] L. Zhang et al., Sparse representation or collaborative representation: which helps face recognition? ICCV, 2011

CRC_RLS

- Key points of CRC_RLS
  - It is the collaborative representation, not the l_1-norm minimization that makes the SRC method works well for face recognition
  - Thus, the l_1-norm regularization can be relaxed to l_2-norm regularization

CRC_RLS

SRC method:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \qquad (1)$$

CRC_RLS:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \qquad (2)$$

(strictly convex)

(1) is not easy to solve; can be solved by iteration methods

However, (2) has a closed-form solution

$$\mathbf{x}_0 = \left(\mathbf{A}^T\mathbf{A} + \lambda E\right)^{-1} \mathbf{A}^T \mathbf{y}$$

can be pre-computed

Can you work it out?

$(\mathbf{A}^{\mathrm{T}}\mathbf{A}+\lambda E)$ is actually positive definite

CRC_RLS

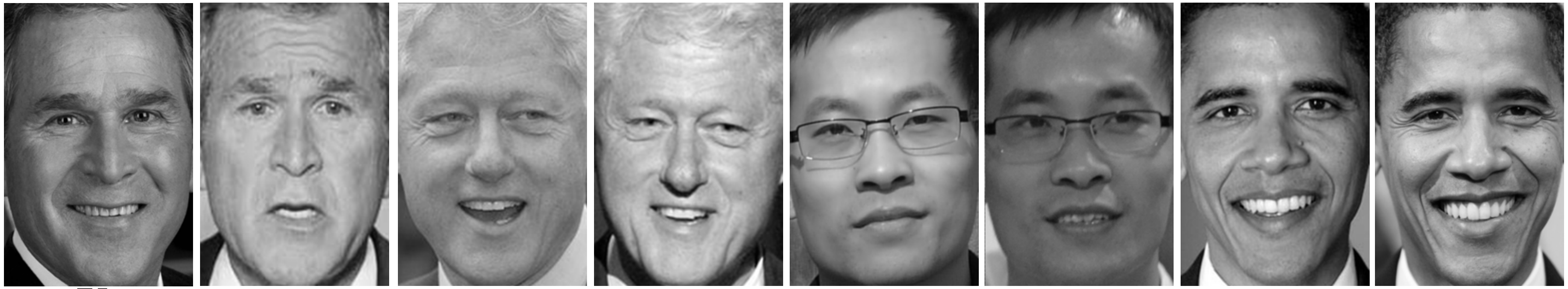
Algorithm

1. **Input**: a matrix of training samples $\mathbf{A} = [A_1, A_2, ..., A_k] \in \mathbb{R}^{m \times n}$ for k classes; $\mathbf{y} \in \mathbb{R}^m$, a test sample;
2. Normalize the columns of $\mathbf{A}$ to have unit l_2-norm
3. Pre-compute $\mathbf{P} = \left(\mathbf{A}^T\mathbf{A} + \lambda E\right)^{-1}\mathbf{A}^T$
4. Code $\mathbf{y}$ over $\mathbf{A}$
$$\mathbf{x}_0 = \mathbf{P}\mathbf{y}$$
5. Compute the residuals $r_i(\mathbf{y}) = \left\|\mathbf{y} - \mathbf{A}\delta_i(\mathbf{x}_0)\right\|_2$, i = {1,...,k}
6. **Output**: $\mathrm{identity}(\mathbf{y}) = \mathrm{argmin}_i r_i(\mathbf{y})$

For $\mathbf{x} \in \mathbb{R}^n$, $\delta_i(\mathbf{x}) \in \mathbb{R}^n$ is a new vector whose only non-zero entries are the entries in $\mathbf{x}$ that are associated with class i

CRC_RLS

- Illustration for CRC_RLS



$\mathbf{v}_{1,1}$ $\mathbf{v}_{1,2}$ $\mathbf{v}_{2,1}$ $\mathbf{v}_{2,2}$ $\mathbf{v}_{3,1}$ $\mathbf{v}_{3,2}$ $\mathbf{v}_{4,1}$ $\mathbf{v}_{4,2}$

$\mathbf{y}$

By solving CRC_RLS,

$$\mathbf{x}_0 = [-0.10, -0.04, -0.09, 0.16, 0.68, 0.14, 0.06, 0.17]^T$$

$$r_1 = \left\| \mathbf{v}_{1,1} \times (-0.10) + \mathbf{v}_{1,2} \times (-0.04) - \mathbf{y} \right\|_2 = 1.14$$

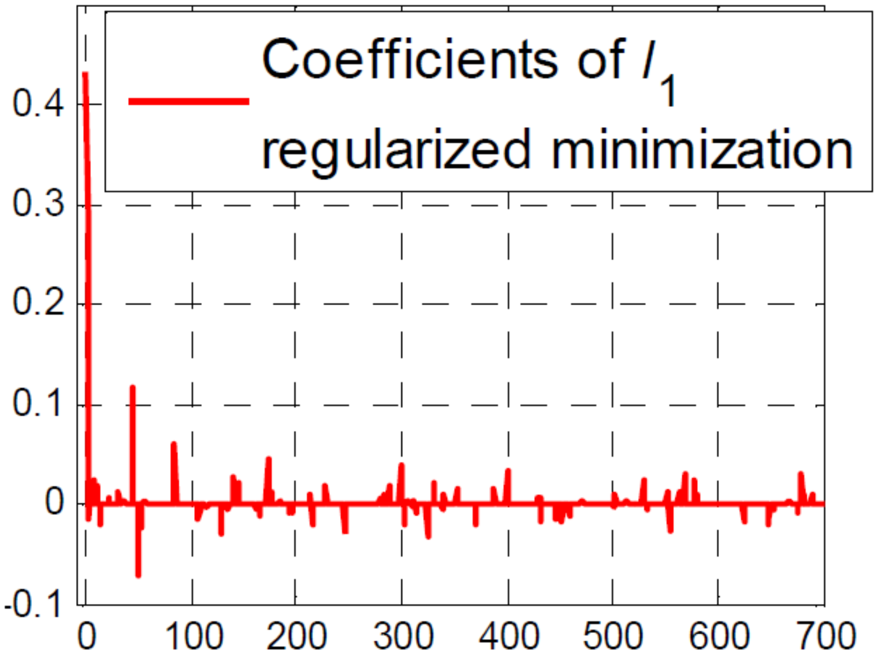
$$r_2 = \left\| \mathbf{v}_{2,1} \times (-0.09) + \mathbf{v}_{2,2} \times (0.16) - \mathbf{y} \right\|_2 = 0.93$$

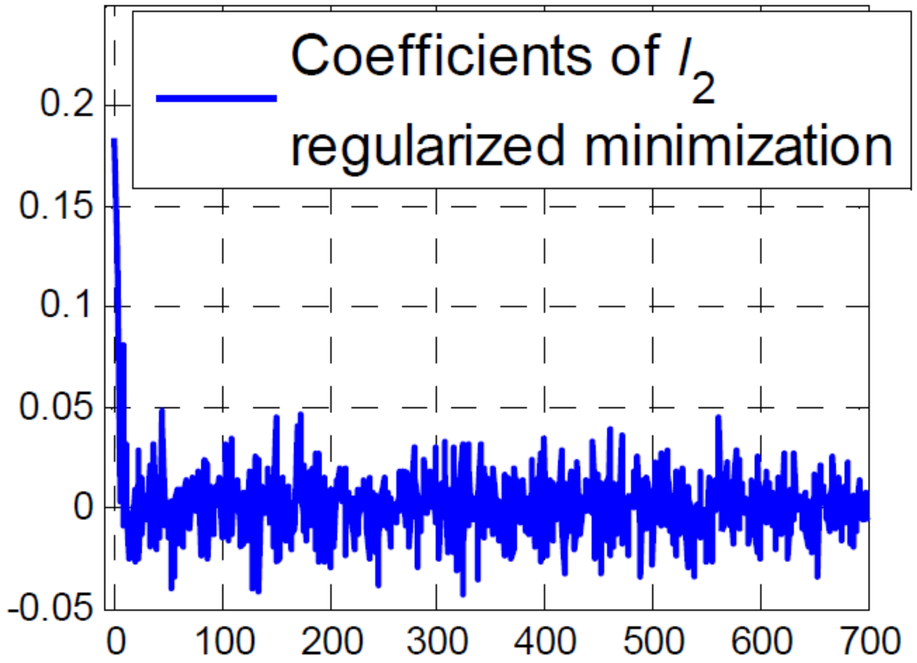
$$r_3 = \left\| \mathbf{v}_{3,1} \times (0.68) + \mathbf{v}_{3,2} \times (0.14) - \mathbf{y} \right\|_2 = 0.27$$

$$r_4 = \left\| \mathbf{v}_{4,1} \times (0.06) + \mathbf{v}_{4,2} \times (0.17) - \mathbf{y} \right\|_2 = 0.79$$

CRC_RLS

- CRC_RLS vs. SRC





The coding coefficients of a query sample

CRC_RLS

- CRC_RLS vs. SRC

	Recognition rate	Time
SRC(l_1_ls)	**0.979**	5.3988 s
SRC(ALM)	**0.979**	0.128 s
SRC(FISTA)	0.914	0.1567 s
SRC(Homotopy)	0.945	0.0279 s
CRC_RLS	**0.979**	**0.0033 s**
Speed-up	**8.5 ~ 1636 times**	

Recognition rate and speed on the Extended Yale B database


Q+A