Lecture 4
Sparse Representation based Classification

Lin ZHANG, PhD
School of Software Engineering
Tongji University
Fall 2020
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

Sparse representation based approach

• Illustration

\[ y = \alpha_{1,1} v_{1,1} + \alpha_{1,2} v_{1,2} + \alpha_{2,1} v_{2,1} + \alpha_{2,2} v_{2,2} \]
\[ + \alpha_{3,1} v_{3,1} + \alpha_{3,2} v_{3,2} + \alpha_{4,1} v_{4,1} + \alpha_{4,2} v_{4,2} \]

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 $A$ for the $n$ training samples of all $k$ object classes

$$A = \begin{bmatrix} A_1, A_2, \ldots, A_k \end{bmatrix} = \begin{bmatrix} v_{1,1}, v_{1,2}, \ldots, v_{k,n_k} \end{bmatrix}$$

Then, the linear representation of a testing sample $y$ can be expressed as

$$y = Ax_0$$

where $x_0 = \begin{bmatrix} 0, \ldots, 0, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}, 0, \ldots, 0 \end{bmatrix}^T \in \mathbb{R}^n$ is a coefficient vector whose entries are zero except those associated with the $i$th class.
Sparse representation based approach

This motivates us to seek the most sparsest solution to $y = Ax$, solving the following optimization problem:

$$x_0 = \arg \min \|x\|_0, \text{s.t.,} \|Ax - y\|_2 \leq \varepsilon \quad (1)$$

where $\|\cdot\|_0$ 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 $x_0$ is sparse enough, the solution of $l_0$-minimization problem is equal to the solution to the following $l_1$-norm minimization problem:

$$x_0 = \arg \min \|x\|_1, s.t., \|Ax - y\|_2 \leq \varepsilon \quad (1)$$

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

There is an equivalent form for (1)

$$x_0 = \arg \min_x \|y - Ax\|_2^2 + \lambda \|x\|_1, \lambda > 0 \quad (2)$$

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
   \[ A = [A_1, A_2, \ldots, A_k] \in \mathbb{R}^{m \times n} \text{ for } k \text{ classes}; \ y \in \mathbb{R}^m, \text{ a test sample; and an error tolerance } \varepsilon > 0 \]

2. Normalize the columns of \( A \) to have unit \( l_2 \)-norm

3. Solve the \( l_1 \)-minimization problem
   \[ x_0 = \arg \min x \|x\|_1, \text{ s.t., } \|Ax - y\|_2 \leq \varepsilon \]

4. Compute the residuals \( r_i(y) = \|y - A \delta_i(x_0)\|_2, \ i = \{1, \ldots, k\} \)

5. **Output**: \( \text{identity}(y) = \arg \min_i r_i(y) \)

For \( x \in \mathbb{R}^n, \ \delta_i(x) \in \mathbb{R}^n \) is a new vector whose only non-zero entries are the entries in \( x \) that are associated with class \( i \)
A valid test image. Recognition with $12 \times 10$ downsampled images as features. The test image $y$ 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 $r_i(y)$ of a test image of subject 1 with respect to the projected sparse coefficients $\delta_i(x_0)$ 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?

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:

\[ x_0 = \arg \min_x \| y - Ax \|^2_2 + \lambda \| x \|_1 \]  \hspace{1cm} (1)

CRC_RLS:

\[ x_0 = \arg \min_x \| y - Ax \|^2_2 + \lambda \| x \|^2_2 \]  \hspace{1cm} (2)

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

However, (2) has a closed-form solution

\[ x_0 = (A^T A + \lambda E)^{-1} A^T y \]

can be pre-computed

\((A^T A + \lambda E)\) is actually positive definite

Can you work it out?
Algorithm

1. **Input:** a matrix of training samples
   \[ A = [A_1, A_2, ..., A_k] \in \mathbb{R}^{m \times n} \] for \( k \) classes; \( y \in \mathbb{R}^m \), a test sample;
2. Normalize the columns of \( A \) to have unit \( l_2 \)-norm
3. Pre-compute
   \[ P = \left( A^T A + \lambda E \right)^{-1} A^T \]
4. Code \( y \) over \( A \)
   \[ x_0 = Py \]
5. Compute the residuals
   \[ r_i(y) = \| y - A \delta_i(x_0) \|_2, \quad i = \{1, \ldots, k\} \]
6. **Output:** identity \( (y) = \arg \min_i r_i(y) \)

For \( x \in \mathbb{R}^n, \quad \delta_i(x) \in \mathbb{R}^n \) is a new vector whose only non-zero entries are the entries in \( x \) that are associated with class \( i \).
By solving CRC_RLS,
\[ x_0 = [-0.10, -0.04, -0.09, 0.16, 0.68, 0.14, 0.06, 0.17]^T \]
\[
\begin{align*}
    r_1 &= \| v_{1,1} \times (-0.10) + v_{1,2} \times (-0.04) - y \|_2 = 1.14 \\
    r_2 &= \| v_{2,1} \times (-0.09) + v_{2,2} \times (0.16) - y \|_2 = 0.93 \\
    r_3 &= \| v_{3,1} \times (0.68) + v_{3,2} \times (0.14) - y \|_2 = 0.27 \\
    r_4 &= \| v_{4,1} \times (0.06) + v_{4,2} \times (0.17) - y \|_2 = 0.79
\end{align*}
\]
CRC_RLS

- CRC_RLS vs. SRC

The coding coefficients of a query sample
## CRC_RLS

- **CRC_RLS vs. SRC**

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC($l_1$-ls)</td>
<td>0.979</td>
<td>5.3988 s</td>
</tr>
<tr>
<td>SRC(ALM)</td>
<td>0.979</td>
<td>0.128 s</td>
</tr>
<tr>
<td>SRC(FISTA)</td>
<td>0.914</td>
<td>0.1567 s</td>
</tr>
<tr>
<td>SRC(Homotopy)</td>
<td>0.945</td>
<td>0.0279 s</td>
</tr>
<tr>
<td><strong>CRC_RLS</strong></td>
<td><strong>0.979</strong></td>
<td><strong>0.0033 s</strong></td>
</tr>
<tr>
<td><strong>Speed-up</strong></td>
<td><strong>8.5 ~ 1636 times</strong></td>
<td></td>
</tr>
</tbody>
</table>

Recognition rate and speed on the Extended Yale B database