A NOVEL KERNEL COLLABORATIVE REPRESENTATION APPROACH FOR IMAGE CLASSIFICATION

Weiyang Liu, Lijia Lu, Hui Li*, Wei Wang, and Yuexian Zou*

School of Electronic and Computer Engineering, Peking University, P. R. China

ABSTRACT

Sparse representation classification (SRC) plays an important role in pattern recognition. Recently, a more generic method named as collaborative representation classification (CRC) has greatly improved the efficiency of SRC. By taking advantage of recent development of CRC, this paper explores to smoothly apply the kernel technique to further improve its performance and proposes the kernel CRC (KCRC) approach. Tested by multiple databases in experiments, KCRC has shown that it can perfectly classify the data with the same direction distribution with limited complexity, and outperforms CRC, SRC and some other conventional algorithms.

Index Terms— Collaborative Representation, Kernel Technique, Regularized Least Square Algorithm, Augmented Lagrange Multiplier Method, Image Classification

1. INTRODUCTION

Sparsity of signals has been commonly considered as a powerful principle in classification and is of increasing importance in pattern recognition recently in academics. Sparse representation technique has been widely studied in [1–3], partially thanks to the great progress of $l_0$ and $l_1$ minimization methods proposed in [4, 5].

It has been found that most natural images can be sparingly represented by over-complete basises, which inspires SRC [1]. Compared to conventional face recognition techniques, SRC can perform better according to experiments in [1]. Although the improvement of recognition accuracy was considered as the contribution of the sparsity of signals, Zhang et al. [6] have revealed that the collaborative representation (CR) mechanism is the real key to improve the recognition accuracy. Moreover, the necessity of sparsity in face recognition was also questioned in [7] and [8]. In summary, developed from SRC, CRC emphasizes CR mechanism, indicating the sparsity constraint only helps robustness, not accuracy. Experiments in [6] showed that CRC method has close or even better recognition results with far less complexity than SRC.

SRC performs poorly when facing data of the same direction distribution [9], because relatively different direction distributions usually are required for the training samples with different labels in SRC, and CRC also shares the same limitation. Therefore, the kernel function, which was originated from SVM [10, 11] and was successfully applied to KPCA [12] and KFDA [13], is adopted in this paper to overcome such shortcoming. A Mercer kernel implicitly defines a nonlinear mapping which maps data in the input space into a high or even infinite dimensional kernel feature space where features of the same class are more easily grouped together and therefore features of different classes become linear separable. It was shown in [9, 14] that SRC performs better by adopting the kernel function. The contribution of this paper is that we smoothly combine CRC method and the kernel technique to establish a novel kernel CRC approach that has absorbed the advantages of both techniques.

The outline of the paper is as follows. In Section II, CRC and the kernel technique are briefly introduced. In Section III, the proposed KCRC and two specific KCRC algorithms are presented. Experimental results are provided and discussed in Section IV, and concluding remarks are given in Section V.

2. PRELIMINARIES

2.1. Principle of CRC

The principle of CRC [6] is briefly presented in this section. In CRC, the dictionary $D$ is constructed by all training samples and a test sample $y$ is coded collaboratively over the whole dictionary, which is the essence of CRC. Assume there is a set of training samples containing total $k$ classes, the $i$th class with $n_i$ samples. The training data matrix $D$, also called dictionary, can be formulated as

$$D = \{D_1, D_2, \ldots, D_k\} \in \mathbb{R}^{m \times n}$$ (1)

where $n = \sum_{j=1}^{k} n_j$ and $D_i$ denotes the dictionary associated with $i$th class, namely

$$D_i = \{v_{i1}, v_{i2}, \ldots, v_{in_i}\} \in \mathbb{R}^{m \times n_i}$$ (2)

in which $m$ is the dimension of each sample and equals to $w \times h$ in the context of image recognition. $v_{ij}$, also called atom, stands for the $j$th training image in the $i$th class.
According to the representation principle, a test image \( y \) belonging to the \( i \)th class can be represented by
\[
y \approx x_1v_{1i} + x_2v_{2i} + \cdots + x_mv_{mi},
\]
(3)
where \( x_{ij} \in \mathbb{R} \), \( j=1, \ldots, n_i \) is the coding coefficient of vector \( v_{ji} \). By defining the vector \( x = [x_1, \ldots, x_i, \ldots, x_m]^T \) in which \( x_i \) is the coding coefficients associated with \( D_i \), the test sample \( y \) can also be formulated as
\[
y \approx Dx \in \mathbb{R}^m
\]
(4)
where \( \{0, \ldots, 0, x_{i1}, \ldots, x_{im}, 0, \ldots, 0 \} \) is the ideal form of the coefficient vector \( x \) in SRC. Only coefficients of the class to which the test sample belongs are non-zero. However, \( x \) is usually not necessary to be sparse in CRC.

The test sample \( y \) can be written as \( y = \hat{y} + e \) where \( \hat{y} = Dx \) is the estimated test sample and \( e \) is the residual caused by noise, occlusion and etc. The vector \( x \) can be found by solving the constrained optimization problem as follows:
\[
x = \arg \min_x \|x\|_p \text{ subj. to } \|y - Dx\|_q \leq \varepsilon
\]
(5)
where \( \varepsilon \) is a small error constant. After Lagrangian formulation, a general model of CRC method is formulated as
\[
x = \arg \min_x \left( \|y - Dx\|_q + \mu \|x\|_p \right)
\]
(6)
where \( \mu \) is the regularization parameter and \( p, q \in \{1, 2\} \). The combinations of \( p, q \) values lead to different instantiations of CRC model. For instance, SRC method is under the condition of \( p=1, q=1 \) and different settings of \( q \) is used to handle recognition with or without occlusion. Similar to SRC, CRC uses the coding coefficients \( \hat{x} \) to obtain the reconstructed image and determine its class label with following equation:
\[
identity(y) = \arg \min_i \left( \|y - D_i \hat{x}_i\|_2 / \|\hat{x}_i\|_2 \right)
\]
(7)
In fact, the key to reduce the computational complexity is to reasonably set the value of \( p, q \). Based on different combinations of \( p, q \), two CRC algorithms were proposed in [6, 15]. One is the CRC regularized least square (CRC-RLS) algorithm with \( p=2, q=2 \). Another one is the robust CRC (RRC) algorithm with \( p=2, q=1 \). It has been comprehensively analyzed in [6] that sparsity of signals is useful but not crucial for face recognition and has been proven that the collaborative representation mechanism does play an important role.

2.2. Kernel Technique

Kernel function [12, 13], also described as kernel technique, has become a simple yet useful tool in pattern recognition since its success in SVM [10, 11], KPCA [12] and KFDA [13]. Kernel function is used to create a nonlinear mapping mechanism \( y \in \mathbb{R} \to \phi(y) \in \mathbb{H} \) where \( \mathbb{H} \) is a unique associated reproducing kernel Hilbert space. If every data point is mapped into higher dimensional space via transformation \( \phi \), the inner product, which is called kernel function, becomes
\[
K(y_i, y_j) = \langle \phi(y_i), \phi(y_j) \rangle = \phi(y_i)^T \phi(y_j)
\]
(8)
where \( y_i, y_j \) are different samples and \( \phi \) denotes the implicit nonlinear mapping associated with the kernel function \( K(y_i, y_j) \). There are some empirical frequently-used kernel functions satisfying the Mercer condition such as the linear kernel \( K(y_i, y_j) = y_i^T y_j \) and Gaussian radial basis function (RBF) kernels \( K(y_i, y_j) = \exp(-\beta\|y_i - y_j\|^2_2) \).

3. PROPOSED KCRC APPROACH

3.1. Formulation of KCRC

It is well-known that data nonlinearly transformed to higher dimensional space is more separable. It could also avoid the same direction distribution of data. Obviously, the new features in kernel space also can be applied to CRC. However, mapping to high dimensional space makes optimization problem in CRC more complicated so that the dimensionality in the kernel feature space has to be reduced. The nonlinear mapping mechanism is formulated as
\[
y \in \mathbb{R}^m \mapsto \phi(y) = [\phi_1(y), \phi_2(y), \ldots, \phi_n(y)] \in \mathbb{R}^s
\]
(9)
where \( \phi(y) \) is the high dimensional feature associated to the sample \( y \) and \( s \gg m \). Define that \( D_{ij} \) denotes the \( i \)th atom in \( D \) with the \( j \)th class label, in which \( 1 \leq i \leq n = \sum_{j=1}^{k} n_j \), but the class label \( [j] \) will be removed for the convenience of description. According to the nonlinear mapping mechanism, the original dictionary \( D \) becomes a much higher dimensional one: \( \Phi = [\phi(d_1), \phi(d_2), \ldots, \phi(d_n)] \in \mathbb{R}^{s \times n} \), and the test sample becomes \( \phi(y) = \Phi x \). The model of KCRC can be formulated as
\[
x = \arg \min_x \|x\|_p \text{ subj. to } \|\phi(y) - \Phi x\|_q \leq \varepsilon.
\]
(10)
However, Eq. (10) is even harder to solve than the previous problem Eq. (5) because the high dimensionality leads to high complexity.

A dimensionality reduction matrix \( R \), namely a projection matrix, can be defined or constructed by utilizing the methodology of KPCA [12] and KFDA [13]. With the matrix \( R \in \mathbb{R}^{s \times c} \), we could obtain
\[
R^T \phi(y) = R^T \Phi x.
\]
(11)
where \( R \) is related to the samples in the kernel space. In fact, each atom in \( R \) is a linear combination of samples in kernel feature space. Namely
\[
R = \Phi \Psi = \{\phi(d_1), \ldots, \phi(d_n)\} \cdot \{\psi_1, \ldots, \psi_c\}
\]
(12)
where \( R = \{R_1, \ldots, R_c\} \) and \( \psi_i \) is the \( n \)-dimensional linear projection coefficients vector corresponding to the \( R_i = \)
The test sample is transformed to $\Psi^TK(D, y)$. For high level corruption and occlusion, kernel robust CRC (KCRC) algorithm ($p=2$, $q=1$) is designed. $D''=\Psi^T G$ is the new dictionary and $P_k''$ is the coding basis in the kernel space.

### 3.2. Proposed KCRC Algorithms

Two algorithms are designed for KCRC. For normal situations, $p$, $q$ are both set as 2. The regularized least square algorithm is applied to solve the model with $p$, $q=2$. Specifically, we give the new dictionary $D'='\Psi^T G$ and define $P'$ as the coding basis in kernel CRC-RLS (KCRC-RLS). Namely

$$P' = \left((\Psi^T G)^T (\Psi^T G) + \mu I\right)^{-1} (\Psi^T G)^T.$$  

The test sample is transformed to $\Psi^T K(D, y)$. For high level corruption and occlusion, kernel robust CRC (KCRC-C) algorithm ($p=2$, $q=1$) is designed. $D''=\Psi^T G$ is the new dictionary and $P_k''$ is the coding basis in the kernel space.

### 4. EXPERIMENTS AND RESULTS

#### 4.1. Experiments on Data Sets with the Same Direction Distribution

Comparisons between CRC and KCRC on self-built data sets are presented. Two class $p$-dimensional data sets are built: $Q=\{q_1, \ldots, q_n\} \in \mathbb{R}^{p \times n}$, $W=\{w_1, \ldots, w_n\} \in \mathbb{R}^{p \times n}$, where $q_i$, $w_i \in [1, 3]$ and $q_i=-q_{i+1}$, $w_j=-w_{j+1}$. Then we uniformly pick value from $[1, 3]$ and $[-3, -1]$ to construct $q_1$, $w_1$ respectively. Then all data is corrupted by Gaussian noise with zero mean and 0.04 variance. By experiments, 2-dimensional situation is shown in Figure 1(a) and Figure 1(b). With different dimension $p=2, 8, 16, 32, 64, 128, 256$, recognition rates of CRC and KCRC are shown in Figure 1(c). It can be found out that KCRC-RLS performs much better than CRC-RLS when facing data with the same direction distribution.

#### 4.2. Experiments on Face Data Sets

To evaluate the performance of the proposed KCRC algorithms, a public face data set (Extended YaleB database) has been used. The specific experiment setup is the same as [1].
Recognition result on self-built data set. In (a) and (b), “+” represents data set $Q$ while “◦” stands for data set $W$. Gray area and white area are judged to class $Q$ and $W$ respectively. (a) Decision boundary by CRC-RLS. (b) Decision boundary by KCRC-RLS. (c) Recognition rate vs. dimension.

4.2.1. Recognition Performance Test

The recognition performance test has used random test [1] based on Extended Yale B database. Each result is averaged out by 20 times experiments. Recognition accuracy of KCRC, CRC, SRC and linear SVM are compared in experiments shown in Figure 2, in which we use the random projection matrix as our $\Psi$. Moreover, downsample is adopted for face feature extraction. It is obtained that KCRC-RLS can achieve the best recognition accuracy only with slightly more running time than CRC-RLS, but much less than SRC.

4.2.2. Robustness Test

Since KCRC-RLS algorithm is developed to cope with the recognition task under low occlusion and corruption, KRCRC algorithm is designed to process the recognition with high occlusion and corruption. Pixel corruption and random block occlusion are adopted to do robustness test whose results are shown in Table 1. Experiments show the robustness of KRCRC is slightly better than RCRC. So KRCRC preserves good robustness.

### 4.3. Experiments on Car Logos Data Sets

This experiment is to further test the performance of KRC approach. A portion of car logos are automatically collected by a video surveillance camera while others are downloaded from the internet. So the car logo images have large variations in terms of image resolution, lighting condition, noise condition, background et al. Data samples are given in Figure 3(a). There are 3 data sets: 300 images (10 classes), 500 images (13 classes) and 1000 images (21 classes). We randomly select 10 images in each class as training samples and the remaining as test samples. Results are averaged across 20 times experiments for accuracy. It can be seen that KCRC-RLS and KRCRC perform well in the experiment. Stable recognition rates show that KRCRC is more robust than KCRC-RLS.

<table>
<thead>
<tr>
<th>Oclusion</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-SRC(BP)</td>
<td>100%</td>
<td>100%</td>
<td>99.3%</td>
<td>97.9%</td>
<td>89.7%</td>
<td>63.9%</td>
</tr>
<tr>
<td>RCRC</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97.3%</td>
<td>92.5%</td>
<td>83.1%</td>
</tr>
<tr>
<td>KRCRC</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>97.8%</td>
<td>93.1%</td>
<td>83.7%</td>
</tr>
</tbody>
</table>

4.3. Experiments on Car Logos Data Sets

This experiment is to further test the performance of KCRC approach. A portion of car logos are automatically collected by a video surveillance camera while others are downloaded from the internet. So the car logo images have large variations in terms of image resolution, lighting condition, noise condition, background et al. Data samples are given in Figure 3(a). There are 3 data sets: 300 images (10 classes), 500 images (13 classes) and 1000 images (21 classes). We randomly select 10 images in each class as training samples and the remaining as test samples. Results are averaged across 20 times experiments for accuracy. It can be seen that KCRC-RLS and KRCRC perform well in the experiment. Stable recognition rates show that KRCRC is more robust than KCRC-RLS.

<table>
<thead>
<tr>
<th>Corruption</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-SRC(BP)</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>91.0%</td>
<td>6.9%</td>
</tr>
<tr>
<td>RCRC</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>91.1%</td>
<td>18.2%</td>
</tr>
<tr>
<td>KRCRC</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>91.9%</td>
<td>20.2%</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this work, we have elaborated the CRC method [6] and proposed the kernel CRC approach. Kernel technique is smoothly combined with the CRC method. The proposed KRCR enhances the discrimination ability of CRC, making the decision boundary more reasonable, and achieves the high recognition accuracy with low complexity. Experiments verified that KRCRC approach outperforms CRC, SRC and linear SVM.
6. REFERENCES


