Discriminative Collaborative Representation for Classification

Yang Wu[1], Wei Li[2], Masayuki Mukunoki[1], Michihiko Minoh[1], and Shihong Lao[3]

[1] Academic Center for Computing and Media Studies, Kyoto University, Kyoto 606-8501,Japan
[2] Institute of Scientific and Industrial Research, Osaka University, Ibaraki-shi 567-0047, Japan
[3] OMRON Social Solutions Co., LTD, Kyoto 619-0283, Japan
yangwu@mm.media.kyoto-u.ac.jp, seuliwei@126.com,
{minoh,mukunoki}@media.kyoto-u.ac.jp, lao_shihong@oss.omron.co.jp

Abstract. The recently proposed l_2-norm based collaborative representation for classification (CRC) model has shown inspiring performance on face recognition after the success of its predecessor — the l_1-norm based sparse representation for classification (SRC) model. Though CRC is much faster than SRC as it has a closed-form solution, it may have the same weakness as SRC, i.e., relying on a "good" (properly controlled) training dataset for serving as its dictionary. Such a weakness limits the usage of CRC in real applications because the quality requirement is not easy to verify in practice. Inspired by the encouraging progress on dictionary learning for sparse representation, which can much alleviate this problem, we propose the discriminative collaborative representation (DCR) model. It has a novel classification model well fitting its discriminative learning model. As a result, DCR has the same advantage of being efficient as CRC, while at the same time showing even stronger discriminative power than existing dictionary learning methods. Extensive experiments on nine widely used benchmark datasets for both controlled and uncontrolled classification tasks demonstrate its consistent effectiveness and efficiency.

1 Introduction

Sparse representation based classification (SRC) [1] has recently attracted a lot of attention due to its simplicity and striking performance on some visual classification tasks especially face recognition. While most followers have focused on exploring new applications or designing new dictionary learning models to further improve its performance on those exemplary recognition tasks, there are also a few papers on exposing the intrinsic reasons for its effectiveness or even expressing different opinions. Among these voices, there is a distinctive argument: it is the collaborative representation of the test sample using all the training samples that truly results in SRC's success but not its l_1-norm based sparsity [2]. To prove that, a new model named collaborative representation based classification (CRC) has been proposed which uses the l_2-norm based regularization to replace the l_1-norm based sparsity term. Primary experiments on face recognition have shown that CRC performs no worse than SRC. However, considering that SRC requires a controlled training set with sufficient samples per class for ensuring a good performance [3], it is likely that CRC may need similar pre-conditions since both of them directly use the training data as the reconstruction dictionary.

To alleviate the dependence on the quality of training data, great efforts have been put into dictionary learning (DL) models [4] [5] [6] for enhancing SRC. They generally aim at learning a dictionary and/or classification model for better exploring the discriminative ability of the training data. Existing DL approaches are very diverse in model design and optimization, resulting in different performances and speeds. As far as we are aware, these approaches are all proposed for sparse representation, and the l_0-norm or l_1-norm based sparsity usually leads to a high computational cost. Since the efficient l_2-norm based collaborative representation has already shown some of its discriminative power, it is interesting and valuable to see whether DL can also be explored to further improve its performance while at the same time keep being efficient. This study is planned for presenting the first attempt in this direction.

More concretely, we propose a novel dictionary learning model called discriminative collaborative representation (DCR), which has stronger discriminative power than state-of-the-art DL models while at the same time utilizes the efficient l_2-norm to regularize the representation coefficients. In addition to that, a novel classification model directly derived from the learning model is adopted. We will show that DCR learns faster and performs better than its competitors on various classification tasks.

2 Related work

Dictionary learning has recently become an active research topic. Though it has been used for many applications, we are focusing on classification tasks.

As its name shows, dictionary learning approaches usually directly target at learning a discriminative dictionary. A representative work is the meta-face learning approach [7] which learns class-specific sub-dictionaries independently. Later on, the DLSI model [8] was proposed to improve the discrimination ability of the sub-dictionaries and also explore their shared common bases via exploring the incoherence between the sub-dictionaries. Very recently, a new model called DL-COPAR [6] develops DLSI's idea on exploring the common bases of sub-dictionaries [8] by explicitly separating the particularity (class-specific sub-dictionaries) and commonality (a common sub-dictionary) in dictionary learning.

There are also some other approaches working in the direction of learning a discriminative classification model using the sparse representation coefficients. Representative approaches include supervised dictionary learning [9] using the logistic regression model, discriminative K-SVD (D-KSVD) [10] with a linear regression model, and the label consistent K-SVD (LC-KSVD) model [4] which adds one more linear regression term to D-KSVD to further enhance the label consistency within each class.

Taking into account the effectiveness of both directions, the work of Fisher discrimination dictionary learning (FDDL) [5] explicitly combines discriminative dictionary learning and coefficients based classification model learning, and uses both of them in its two classification models as well.

The proposed DCR, however, integrates the key ideas behind all these three groups and uses l_2-norm regularization terms for efficiency while pursuing effectiveness and comprehensiveness. Moreover, DCR has a novel classification model which coincides well with its learning model. The model optimization with closed-form solutions for

alternating steps and the comprehensiveness of experiments in this paper are also different from those in the literature.

It's worth noticing that the work of Discriminative k-metrics [11] extends q-flats to metrics and introduces discrimination, resulting in a similar formula with part of DCR. However, its collaborative representation (CR) exists only in within-class metrics, while DCR is an inter-class CR model. The work on structured sparsity [12] also has a CR-like model, but it embeds GMM/MAP-EM for representation and uses PCA to generate dictionary and regularize coefficients, which are much unlike DCR.

3 Discriminative Collaborative Representation

3.1 Sparse/Collaborative representation

Given a training dataset $X = [X_1, \ldots, X_L] \in \mathbb{R}^{d\times n}$, where n denotes the total number of samples, d denotes their feature dimension, L is the number of classes, and $X_i, \forall i \in \{1, \ldots, L\}$ denotes the n_i samples belonging to class i. Both sparse representation and collaborative representation seeks a linear combination of all the training samples X to best reconstruct an arbitrary test sample $\mathbf{y} \in \mathbb{R}^d$. And such a reconstruction is regularized by some norm of the reconstruction coefficients to make the solution unique. It can be modeled by the following optimization problem:

$$\hat{\alpha} = \arg\min_{\alpha} \ \|\mathbf{y} - X\alpha\|_2^2 + \lambda_1 \|\alpha\|_p, \tag{1}$$

where λ_1 is a trade-off parameter for balancing the reconstruction error and the squared norm of α. In general, the l_p-norm can be any feasible norm. Since l_0-norm leads to a combinatorial optimization problem which is hard to be solved efficiently, SRC chooses the l_1-norm which to some extent ensures the sparsity of α, coinciding with the belief that the sparse coefficients have great discriminative ability. Differently, CRC takes the l_2-norm (actually the squared l_2-norm is used for easier optimization) which cannot make α sparse any more, but it leads to an efficient closed-form solution with good classification performance [2] as well.

3.2 Dictionary learning for DCR

Reconstruction using the training data itself makes the performance of SRC and CRC largely depend on the properties of the training data X. To alleviate such a dependance, there is a research direction of learning a better dictionary D from X to replace it for the reconstruction. Inspired by the existing dictionary learning (DL) approaches, we design our dictionary learning model for DCR as:

$$\langle D^*, W^*, T^*, A^* \rangle = \arg\min_{D,W,T,A} \left\{ r\left(X, D, A\right) + \lambda \left(\|A\|_F^2 + \|D\|_F^2 \right) + \gamma f\left(W, T, A\right) \right\}, \tag{2}$$

where $D \in \mathbb{R}^{d\times K}$ with K items is the learned dictionary from X (usually $K \leq n$); $A \in \mathbb{R}^{K\times n}$ denotes the reconstruction coefficients over D for all the n training samples; W and T denote the learned parameters of the discriminative model $f(W, T, A)$ for

classification with A; $r(X, D, A)$ is the discriminative reconstruction model defined over D (called the discriminative fidelity in [5]); $\|\cdot\|_F$ denotes the Frobenius norm which is a generalization of l_2-norm from dealing with vectors to operating on matrices. λ and γ are two unavoidable trade-off parameters, for which the discussion will be given in section 4.

Most of the existing DL models can be covered by the above general model if we replace the Frobenius norm with a sum of l_0-norms or l_1-norms (except some models do not have the $f(W, T, A)$ term). The differences of them mainly consist in their detailed design of $r(X, D, A)$, $f(W, T, A)$ and the regularization of D, which largely influences the performance and speed of the model. Different from existing models, we propose to use the following formula for $r(X, D, A)$ in our model:

$$r\left(X, D, A\right) = \left\|X - DA\right\|_F^2 + \sum_{i=1}^{L} \left\|X_i - D_i A_i^i - D_0 A_i^0\right\|_F^2 + \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left\|D_i A_j^i\right\|_F^2. \quad (3)$$

In this formula, $D = [D_0, D_1, \ldots, D_L]$ denotes the dictionary to be learned, where D_0 is a common sub-dictionary shared by all the classes while D_i with $i \in \{1, \ldots, L\}$ stands for a class-specific sub-dictionary. Accordingly, A_j^i denotes the coefficients corresponding to the sub-dictionary D_i for those samples from X_j (i.e. the columns of A_j^i correspond to class j). When the physical meanings are concerned, the first term is the *global reconstruction error* (ensuring that the whole dictionary D can well represent X); the second term is the *class-specific reconstruction error* (forcing D_i together with D_0 to be able to well represent X_i); and the third term is the *confusion factor* (restricting D_i's ability on reconstructing samples from any other classes rather than i). Please note that the first term and the second term are not the same. The second term doesn't count $D_j, \forall j \neq i$. Putting them together is to force D discriminative.

For $f(W, T, A)$, we use the same discriminative model as the one for LC-KSVD [4] (more precisely the LC-KSVD2 model in the original paper)[4]:

$$f\left(W, T, A\right) = 4\left\|Q - TA\right\|_F^2 + \left\|H - WA\right\|_F^2, \quad (4)$$

where $H = [\mathbf{h}_1, ..., \mathbf{h}_n] \in \mathbb{R}^{L\times n}$ are label vectors for X with $\mathbf{h}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^L$ indicating which class $\mathbf{x}_i$ is belonging to, and $Q = [\mathbf{q}_1, ..., \mathbf{q}_n] \in \mathbb{R}^{K\times n}$ are ideal sparse codes for X with $\mathbf{q}_i = [0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0]^T \in \mathbb{R}^K$ in which only the items corresponding to D_k is 1 when $\mathbf{x}_i$ is belonging to class k. In fact, given the structure of D, Q can be directly derived from H. The discriminative model aims at learning a linear mapping T which can map the coefficients A to the desired Q, while at the same time learning a linear regression model W which can transfer A to its corresponding label vectors. Therefore, W can be viewed as the model parameters of a linear classifier, while T acts like a parter of W which has greater modeling ability (with more parameters) than it. Such a design has been proved to be very effective in LC-KSVD, and it is more efficient than the Fisher discriminant based discriminative model in FDDL.

[4] We follow LC-KSVD on balancing the two parts with a factor of 4 for simplicity, though a better factor may exist.

3.3 Optimization

Like other DL models, the optimization can only be done by alteratively optimize model parameters (D, T, and W) and coefficients A until convergence.

Initialization D and A We simply utilize principle component analysis (PCA) to initialize D_0 and $D_i, \forall i \in \{1, \ldots, L\}$ with X and X_i, respectively. However, it is also possible to initialize D with random numbers, which will only cost a few more optimization iterations.

Unfortunately, there is no plausible way to initialize T and W without knowing A, while initializing A also needs some existing T and W. Therefore, we choose to discard $f(W, T, A)$ at first, so that A can be initialized based only on an initial D. More concretely, A can be computed in a class-by-class way thanks to the decomposition ability of the Frobenius norm. In another word, for each $i \in \{1, \ldots, L\}$, A_i can be initialized independently as follows.

$$A_i^* = \arg\min_{A_i} \left\{ \|X_i - DA_i\|_F^2 + \left\|X_i - DS_{0i}S_{0i}^T A_i\right\|_F^2 + \left\|DS_{\backslash 0i}S_{\backslash 0i}^T A_i\right\|_F^2 + \lambda \|A_i\|_F^2 \right\}, \tag{5}$$

where

$$\begin{aligned} S_i &= \begin{bmatrix} \mathrm{O}_{\sum_{m=1}^{i-1} K_m \times K_i} \\ I_{K_i \times K_i} \\ \mathrm{O}_{\sum_{m=i+1}^{L} K_m \times K_i} \end{bmatrix}, \forall i \in \{0, 1, \ldots, L\}, \\ S_{0i} &= [S_0, S_i], \\ S_{\backslash 0i} &= [S_1, \cdots, S_{i-1}, S_{i+1}, \cdots, S_L], \end{aligned} \tag{6}$$

with $K_i, i \in \{0, 1, \ldots, L\}$ denoting the dictionary size of D_i. Here S_{0i} is a matrix for selecting D_0 and D_i, while $S_{\backslash 0i}$ is for discarding D_0 and D_i. O and I denote the zero matrix and the identity matrix, respectively. The optimization problem of Equation 5 can be rewritten into a simpler form

$$A_i^* = \arg\min_{A_i} \left\{ \|R_i - Z_i A_i\|_F^2 + \lambda \|A_i\|_F^2 \right\}, \tag{7}$$

where

$$R_i = \begin{bmatrix} X_i \\ X_i \\ \mathrm{O}_{d \times n_i} \end{bmatrix}, Z_i = \begin{bmatrix} D \\ DS_{0i}S_{0i}^T \\ DS_{\backslash 0i}S_{\backslash 0i}^T \end{bmatrix}. \tag{8}$$

Therefore, A_i has a computationally very efficient closed-form solution just like the CRC model:

$$A_i^* = \left(Z_i^T Z_i + \lambda \cdot I\right)^{-1} Z_i^T R_i. \tag{9}$$

Optimizing D, T, and W when given A Once A is given, the term $\lambda \|A\|_F^2$ becomes a constant, however, D is still impossible to be optimized as a whole because the objective function in Equation 2 has two terms which are functions of sub-dictionaries $D_i, i \in \{0, 1, \ldots, L\}$ but not the overall dictionary D. Therefore, we optimize D_is one-by-one, assuming the others are fixed.

First, for each **class-specific sub-dictionary** $D_i, i \in \{1, \ldots, L\}$, suppose all $D_j, j \in \{0, \ldots, L\}, j \neq i$ are fixed, we can reform the objective function for optimizing D_i as

$$D_i^* = \arg\min_{D_i} \left\{ \|U_i - D_i V_i\|_F^2 + \lambda \|D_i\|_F^2 \right\}, \tag{10}$$

where

$$U_i = \left[X - D_{\setminus i} A^{\setminus i}, (X_i - D_0 A_i^0), \mathrm{O}_{d \times (n-n_i)} \right], \quad V_i = \left[A^i, A_i^i, A_{\setminus i}^i \right]. \tag{11}$$

In Equation 11, $D_{\setminus i}$ denotes all the $D_j, j \in \{0, \ldots, L\}, j \neq i$ together and $A^{\setminus i}$ denotes their corresponding coefficients (i.e. without A^i). Conceptually, "$\setminus i$" means without class i. $\mathrm{O}_{d \times (n-n_i)}$ denotes a $d \times (n - n_i)$ dimensional zero matrix, where $n_i, i \in \{1, \ldots, L\}$ is the number of samples in class i and $n = \sum_{i=1}^{L} n_i$. Equation 10 has a closed-form solution

$$D_i^* = U_i V_i^T \left(V_i V_i^T + \lambda \cdot I \right)^{-1}. \tag{12}$$

Then, for the **common sub-dictionary** D_0, when $D_i, i \in \{1, \ldots, L\}$ are all given, the optimizing objective function for D_0 can also be reformed as

$$D_0^* = \arg\min_{D_0} \left\{ \|U_0 - D_0 V_0\|_F^2 + \lambda \|D_0\|_F^2 \right\}, \tag{13}$$

where

$$U_0 = \left[X - D_{\setminus 0} A^{\setminus 0}, (X - D_{\setminus 0} \hat{A}^{\setminus 0}) \right], \quad V_0 = \left[A^0, A^0 \right], \tag{14}$$

with

$$\hat{A}^{\setminus 0} = \begin{bmatrix} A_1^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & A_2^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & A_L^L \end{bmatrix}. \tag{15}$$

Similarly, Equation 13 has a closed-form solution

$$D_0^* = U_0 V_0^T \left(V_0 V_0^T + \lambda \cdot I \right)^{-1}. \tag{16}$$

After getting A and D, optimizing T becomes solving a simple linear regression problem:

$$T^* = \arg\min_{T} \|Q - TA\|_F^2, \tag{17}$$

whose solution is $T^* = QA^T(AA^T)^{-1}$. Similarly, W also has a closed-form solution $W^* = HA^T(AA^T)^{-1}$.

Note that optimizing D_i depends on a given $D_{\setminus i}$, therefore, once D_i is updated, it should be used to update each $D_j, j \neq i$ in $D_{\setminus i}$. This is a chicken-and-egg problem, so a straightforward solution is updating all the D_is iteratively until they are converged (i.e. getting very small changes). However, since we are iterating between optimizing A and updating D, T, and W, a converged D will soon been changed once A is recomputed. Therefore, in our implementation, we ignored the iteration in D's optimization, and found that it still worked very well for our experiments.

Optimizing A when given D, T, and W Similar to initializing A, when D, T, and W are given, optimizing A is equivalent to optimizing A_i for each $i \in \{1, \ldots, L\}$ independently as follows.

$$A_i^* = \arg\min_{A_i} \left\{ \begin{array}{l} \|X_i - DA_i\|_F^2 + \left\|X_i - DS_{0i}S_{0i}^T A_i\right\|_F^2 + \left\|DS_{\setminus 0i}S_{\setminus 0i}^T A_i\right\|_F^2 \\ +\lambda \|A_i\|_F^2 + \gamma \left(4 \|Q_i - TA_i\|_F^2 + \|H_i - WA_i\|_F^2\right) \end{array} \right\}, \quad (18)$$

which is very much like Equation 5 except the last two additional terms, and it can be rewritten into Equation 7 as well, leading to the same solution as Equation 9. The differences are the values of R_i and Z_i, which now contain two extra terms about T and W as follows.

$$R_i = \begin{bmatrix} X_i \\ X_i \\ \mathrm{O}_{d \times n_i} \\ 2\sqrt{\gamma}Q_i \\ \sqrt{\gamma}H_i \end{bmatrix}, Z_i = \begin{bmatrix} D \\ DS_{0i}S_{0i}^T \\ DS_{\setminus 0i}S_{\setminus 0i}^T \\ 2\sqrt{\gamma}T \\ \sqrt{\gamma}W \end{bmatrix}. \quad (19)$$

3.4 Classification model

After learning D, T, and W from training samples, we can use them for classifying an input test sample $\mathbf{y}$, or a set of test samples Y for set-based classification. Both tasks will be verified in our experiments. For simplicity, we use Y to stand for both cases, i.e., Y can be a feature vector for single sample or a matrix whose columns are individual samples belonging to the same set.

Unlike the classification models in the existing approaches, our classification model for DCR directly coincides with its dictionary learning model. For each candidate class $i \in \{1, \ldots, L\}$, suppose Y belongs to class i, then we can compute A_i according to:

$$A_i^* = \arg\min_{A_i} \left\{ \begin{array}{l} \|Y - DA_i\|_F^2 + \left\|Y - DS_{0i}S_{0i}^T A_i\right\|_F^2 + \left\|DS_{\setminus 0i}S_{\setminus 0i}^T A_i\right\|_F^2 \\ +\lambda \|A_i\|_F^2 + \gamma \left(4 \|Q_i - TA_i\|_F^2 + \|H_i - WA_i\|_F^2\right) \end{array} \right\}, \quad (20)$$

whose solution has exactly the same form as the one for Equation 18. The only change needs to make is replacing X_i with Y. Therefore, we get a collaborative representation error $E_i(Y)$ for class i:

$$\begin{aligned} E_i(Y) = &\|Y - DA_i^*\|_F^2 + \left\|Y - DS_{0i}S_{0i}^T A_i^*\right\|_F^2 + \left\|DS_{\setminus 0i}S_{\setminus 0i}^T A_i^*\right\|_F^2 \\ &+ \lambda \|A_i^*\|_F^2 + \gamma \left(4 \|Q_i - TA_i^*\|_F^2 + \|H_i - WA_i^*\|_F^2\right). \end{aligned} \quad (21)$$

Then Y is classified by

$$C(Y) = \arg\min_i E_i(Y). \quad (22)$$

We do have tried other existing classification models like the linear projection model for LC-KSVD and found that they are not as good as the proposed classification model, which fits the learning model better and looks more reasonable. Detailed comparison is omitted in this paper due to the space limits.

Table 1. Computational complexity of DCR. d, K, L, and n refer to the feature dimensionality, the size of the dictionary, the number of classes and the number of samples, respectively.

Training (Dictionary Learning)	
Operation	**Complexity**
Initializing A	$\mathcal{O}\left(dK^2L + K^3L + dKn\right)$
Optimizing D	$\mathcal{O}\left(dKLn + n\sum_{i=0}^{L} K_i^2 + \sum_{i=0}^{L} K_i^3\right)$
• Computing U_i	$\mathcal{O}\left(dn\sum_{j\neq i} K_j + dK_0n_i\right)$
• Computing D_i^*	$\mathcal{O}\left(dK_in + n\sum_{i=1}^{L} K_i^2 + \sum_{i=1}^{L} K_i^3\right)$
• Computing $U_0\&D_0^*$	$\mathcal{O}\left(dK_0n + nK_0^2 + K_0^3\right)$
Optimizing T	$\mathcal{O}\left(K^2n + K^3\right)$
Optimizing W	$\mathcal{O}\left(KLn + K^2n + K^3+\right)$
Optimizing A	$\mathcal{O}\left((d+K+L)K(KL+n)\right)$
• Computing $Z_i\&A_i^*$	$\mathcal{O}\left((d+K+L)K(K+n_i)\right)$
Sub-total	$\mathcal{O}\left((K+n)dKL + K^3L + K^2n\right)$

Testing (Classifying) each sample	
Operation	**Complexity**
Optimizing A	$\mathcal{O}\left((d+K+L)KL\right)$
Computing $E_i(Y), \forall i$	$\mathcal{O}\left((d+K+L)KL\right)$
Classification	$\mathcal{O}\left(L\right)$
Sub-total	$\mathcal{O}\left((d+K+L)KL\right)$

3.5 Convergence and computational complexity

We present the computational complexity for each component/operation of our dictionary learning model and the classification model in Table 1. Since the components of our models only contain simple matrix operations, these complexity functions can be easily verified by checking the corresponding equations. Note that we have used the assumption $L \ll K$, which is generally true, for simplifying some of them when it is necessary. It has been proved that alternating optimization (AO) globally converges for iteration sequences initialized at arbitrary points and it is locally, q-linearly (faster than linearly) convergent to any local minimizer that satisfies some mild assumptions [13], so the AO algorithms can usually converge very quickly. In our case, DCR always converges within 3 to 8 iterations in all the experiments to be presented.[5] Therefore, the sub-total complexity listed in the training stage, which covers the initialization and a single iteration, can also stand for the complexity of the whole alternative optimization process. It can be seen that DCR scales less than linearly with d, L, and n, but nearly proportionally to K^3. Therefore, when the dictionary size is fixed/predetermined, it scales well with the dimensionality of the data and the number of samples. In the testing stage, classifying a single sample has a reasonable complexity. Note that it is benefited from the fact that $\left(Z_i^T Z_i + \lambda \cdot I\right)^{-1} Z_i^T$ can be pre-computed using the learned model. As it will be shown in the next section, DCR has a significantly more efficient learning model than other related dictionary learning methods, while its classification model is comparable to the best of them in efficiency.

[5] Please refer to the supplementary material for more discussions and experimental results.

4 Experiments on real-world applications

We try DCR on solving various real-world classification problems including face recognition in controlled laboratory environments, uncontrolled person re-identification in a real airport surveillance scenario, texture classification with great scale, viewpoint, and illumination changes, and fine-grained object categorization for differentiating leaf species and food subcategories. For each of these four types of classification problems, we choose two different and representative benchmark datasets for evaluating the performance of DCR, comparing with CRC, SRC, and other related dictionary learning models including FDDL, LC-KSVD, and DL-COPAR. We used the version of SRC embedded in the FDDL code, and implemented CRC by ourselves. Codes for all the other methods were got from their authors. State-of-the-art results on specific datasets from other unrelated methods are also listed for reference. Whenever applicable, we conduct 10 times random training and test data sampling for result averaging.

For a clear overview and comparison of all the experiments and their corresponding results, we briefly introduce each classification problem and the concrete tasks, whilst having the dataset statistics listed together with the classification performance in uniform tables. Representative samples images are given to those less well-known datasets for a better understanding. To be brief and clear, the analysis and discussion of the results is stated in a separate subsection after the individual subsections.

4.1 Experimental settings

The same features and sub-dictionary sizes (for DL models only) have been used for all these models to ensure a relatively fair comparison. Though it is possible that different models may favor different dictionary size settings, it is unaffordable to perform a brute-force best setting search for each of them on every dataset due to its high computational cost. Concretely, we had the sub-dictionary sizes (K_0 and $K_i, i \in \{1, \ldots, L\}$) chosen as follows: $K_0 = 5$ and $K_i = 15$ for the Extended Yale B dataset; $K_0 = 3$ and $K_i = 6$ for the AR dataset; $K_0 = 10$ and $K_i = 23$ for the iLIDS-MA dataset; $K_0 = 4$ and $K_i = 8$ for the iLIDS-AA dataset; $K_0 = 2$ and $K_i = 10$ for the KTH-TIPS dataset; $K_0 = 5$ and $K_i = 15$ for the CUReT dataset; $K_0 = 3$ and $K_i = 10$ for the Swedish Leaf dataset; and $K_0 = 3$ and $K_i = 6$ for the PFID Food dataset. For the comparisons with the ScatNet features on texture classification datasets, we have $K_0 = 2$ and K_i equal to the number of training samples per class for every setting. For all our experiments, we used the same trade-off parameters for DCR: $\lambda = 1.0 \times 10^{-4}$ and $\gamma = 2.5 \times 10^{-7}$. For the other methods, we had the following setting for their trade-off parameters (fixed as well): $\lambda = 1.0 \times 10^{-4}$ for SRC and CRC; $\lambda_1 = 1.0 \times 10^{-4}$, $\lambda_2 = 5.0 \times 10^{-3}$, $\gamma = 0.001$ and $w = 0.05$ for FDDL; $\alpha = 1.0 \times 10^{-6}$, and $\beta = 2.5 \times 10^{-7}$ for LC-KSVD. These parameters were chosen by extensive but not brute-force testing for making the results as good as possible for all the methods, while at the same time made to be consistent across them. It's worth mentioning that DCR's performance is stable w.r.t. a large range of λ (from 10^{-8} to 10^{-2}) and γ (from 0 to 10^{-4}). Details on how the performances change with these parameters are omitted due to the space limit. The other parameters (if exist) for the methods compared with were kept as they are in their original codes.

Table 2. The benchmark datasets used for face recognition, their statistics, and the average recognition accuracy for each compared method. The best results are in bold, while those worth mentioning are marked in italic.

Dataset	Statistics				Performance of Methods (%)					
	Samples (NS)	Classes (NC)	Training Samples (per Class) (NTrS(/NC))	Test Samples (per Class) (NTsS(/NC))	SRC [1]	CRC [2]	FDDL [5]	LC-KSVD [4]	DL-COPAR [6]	DCR
Ext. Yale B	2414	38	half ($\sim$ 32)	half ($\sim$ 32)	95.1	*97.6*	96.8	94.4	92.8	**98.2**
AR	1400	100	700 (7)	700 (7)	89.8	*91.9*	91.7	67.7	69.4	**93.4**

4.2 Experiment details

Controlled face recognition Face recognition, more specifically, controlled face recognition in laboratory environments, has been tested on by almost every sparse/collaborative representation based model. We follow such a tradition and choose two widely-used benchmark datasets for our experiments: the Extended Yale B [14] dataset and the AR [15] dataset. The former one contains illumination and facial expression variations, while the later covers one more variation – disguises changes. The AR dataset used here is the one mentioned in [1] and [5], which is a subset of the original dataset. We use the same 504-dimensional feature representation (generated by random matrix projection) as the one adopted in [4] for Extended Yale B dataset, and the 300-dimensional Eigenfaces for AR dataset. The statistics of the experimental data and the final recognition rates are presented in Table 2.

Uncontrolled person re-identification Person re-identification is a problem of identifying people again when they travel across non-overlapping cameras or reappear in the view of the same camera after disappearing for some time. Though any possible cues can be used for solving it, body appearance is mostly concerned. Since almost all the benchmark datasets were built from data captured in real scenarios without specific environmental settings, it is a good uncontrolled recognition problem which is much unlike the above face recognition problem.

In this paper, we work on the two newly built datasets "iLIDS-MA" and "iLIDS-AA" [16] collected from the i-LIDS video surveillance data captured at an airport. This data was originally released by the Home Office of UK. Both of them contain multiple images for each human individual captured by two non-overlapping cameras (camera 1 and camera 3 in their original setting), and there are large viewpoint changes. The iLIDS-MA dataset has 40 persons with exactly 46 manually cropped images per camera for each person, while the iLIDS-AA dataset contains as many as 100 individuals with totally 10754 images (each individual has 21 to 243 images) collected by an automatic tracking algorithm (thus localization errors and unequal class sizes may exist). For result averaging, we random sample certain amount of images per person (23 for iLIDS-MA, and up to 46 for iLIDS-AA) from each camera for training and test, respectively. Some randomly chosen samples are shown in Figure 1. We use the same 400-dimensional color and texture histograms based features as adopted in [17] for all the methods. Following [18], we perform multiple-shot re-identification (i.e., set-based classification). Therefore, the set-based classification model of DCR is used, while the simple minimum point-wise distance between two sets is adopted for other methods

Fig. 1. Some randomly chosen image examples of iLIDS-MA and iLIDS-AA datasets.

Table 3. The benchmark datasets used for person re-identification, their statistics, and the average recognition rates (at rank 10%) of compared methods. The best results are in bold, while those worth mentioning are marked in italic.

Dataset	Statistics				Performance of Methods (%)						
	NS	NC	NTrS(/NC)	NTsS(/NC)	CSA [18]	SRC	CRC	FDDL	LC-KSVD	DL-COPAR	DCR
iLIDS-MA	3680	40	920 (23)	920 (23)	80.5	77.0	72.3	82.3	82.3	**85.3**	*83.3*
iLIDS-AA	≤9200	100	≤4600 (≤46)	≤4600 (≤46)	51.5	*77.9*	64.2	70.0	73.7	*66.6*	**80.3**

except CSA [18]. Since personal re-identification is commonly treated as a ranking problem and we expect to see the correct match in the top-ranked few candidates, we report the cumulative recognition rate at rank top 10% instead of the rank-1 recognition accuracy. The results are shown in Table 3.

Texture classification Unlike other classification tasks, texture classification is useful for verifying the effectiveness of a classification model on working with the texture cue only. Two representative benchmark datasets: KTH-TIPS with 10 classes and CUReT with 61 classes, are chosen for our experiments because they both have enough samples for each class (satisfying SRC's one precondition). However, these two datasets share the same difficulty of having great within-class variations including illumination, viewpoint and scale changes. We use the PRI-CoLBP$_0$ feature proposed in [19] as the raw feature representation which is designed to be somewhat robust to these variations. By doing so, it is more meaningful to compare our results with the state-of-the-art shown in [19], which was generated by Kernel SVM (KSVM) with a χ^2 kernel. The experimental results are listed in Table 4.

Fine-grained object categorization Fine-grained object categorization concerns the classification of sub-categories, thus it lies in the continuum between basic level categorization and identification of individuals. Though it has not been as popular as those

Table 4. The benchmark datasets used for texture classification, their statistics, and the average recognition accuracy for each compared method. The best results are in bold, while those worth mentioning are marked in italic.

Dataset	Statistics				Performance of Methods (%)									
	NS	NC	NTrS (/NC)	NTsS (/NC)	Zhang et al.[20]	Caputo et al.[21]	KSVM [19]	SVM	SRC	CRC	FDDL	LC-KSVD	DL-COPAR	DCR
KTH-TIPS	810	10	400 (40)	410 (41)	96.1	94.8	98.3	86.1	95.7	*97.1*	*69.9*	88.5	*58.5*	**98.7**
CUReT	5612	61	2806 (46)	2806 (46)	95.3	98.5	98.6	82.9	82.4	*93.3*	*4.9*	93.0	*10.3*	**98.9**

Swedish Leaf Food

Fig. 2. Some representative samples from the Swedish leaf dataset and the Pittsburgh Food Image Dataset, respectively.

two extremes, recently its importance has been rediscovered by the community. We experiment on two specific tasks: identifying leaf species in the popular Swedish leaf dataset, and classifying fast food sub-categories in the subset of 61 food classes from the Pittsburgh Food Image Dataset (PFID) [23]. These two tasks covers the problem of using mainly shape cue and the one which is rich of color, texture and shape information. The PFID dataset is more challenging due to the large within-class variations and possibly different data distributions in the training and test subsets. More concretely, there are 3 different instances in the same food sub-category, which were bought from different chain stores on different days, and each instance has six images taken from different viewpoints. In our experiment, two instances are randomly chosen for training while the other is left for testing, so we had 3 trials for result averaging. Representative samples from these two datasets are shown in Figure 2, and the categorization accuracies can be found in Table 5.

4.3 Result analysis and discussion

All the results shown above clearly demonstrate the effectiveness and robustness of DCR. On seven of the eight datasets, DCR performs the best, exceeding all related models and those methods which represent the state-of-the-art. For only iLIDS-MA dataset, its performance is slightly lower than DL-COPAR, but still higher than all the others. The high scores on texture datasets and the leaf dataset are mainly because the adopted PRI-CoLBP$_0$ features themselves are already very effective (see KSVM's or SVM's performance) and there are plenty of samples per class. Though we have tried our best to use the original codes from the authors for the existing methods (like SRC), there may be slight differences between the results reported in the literature and the ones shown here, which may be due to the usage of different features, different data

Table 5. The benchmark datasets used for fine-grained object categorization, their statistics, and the average recognition accuracy for each compared method. The best results are in bold, while those worth mentioning are marked in italic.

Dataset	Statistics				Performance of Methods (%)								
	NS	NC	NTrS (/NC)	NTsS (/NC)	Spatial PACT [22]	Yang et al. [23]	SVM	SRC	CRC	FDDL	LC-KSVD	DL-COPAR	DCR
Swedish Leaf	1125	15	375 (25)	750 (50)	97.9	N/A	95.0	95.8	*99.1*	92.2	99.0	*43.7*	**99.2**
Food	1098	61	732 (12)	366 (6)	N/A	28.2	18.4	31.1	*34.9*	17.5	22.0	*16.7*	**37.3**

Table 6. The benchmark datasets used for texture classification, their statistics, and the average recognition accuracy for each compared method. The best results are in bold, while those worth mentioning are marked in italic.

Dataset	Statistics				Performance of Methods (%)					
	NS	NC	NTrS (/NC)	NTsS (/NC)	KSVM [19]	SRC	CRC	LC-KSVD	ScatNet [24]	DCR
KTH-TIPS_5	810	10	50 (5)	760 (76)	64.4	19.9	35.7	56.6	70.4	**75.0**
KTH-TIPS_20	810	10	200 (20)	610 (61)	83.3	22.5	50.3	60.4	*94.39*	**94.41**
KTH-TIPS_40	810	10	400 (40)	410 (41)	88.6	24.7	54.1	73.6	*97.7*	**97.8**
UIUC_5	1000	25	125 (5)	815 (35)	33.6	21.3	30.0	47.5	49.5	**57.6**
UIUC_10	1000	25	250 (10)	750 (30)	30.2	24.0	34.8	52.0	60.8	**71.4**
UIUC_20	1000	25	500 (20)	500 (20)	28.5	27.6	41.8	61.6	74.6	**78.4**

caused by random sampling, and possibly slightly different parameter settings. Note that FDDL and DL-COPAR seem to significantly over-fit the training data on the last four datasets, which is even worse than using SRC itself.

Though we've already shown some state-of-the-art results from unrelated methods, there are definitely uncovered ones, especially when different features are used. In order to show the superiority of proposed classifier DCR, we take texture classification as an example to show how it performs comparing with those strongest competitors using the same newly proposed feature ScatNet [24]. We use the latest code of ScatNet from its authors, and have the method proposed in [24] (ScatNet with a linear SVM classifier) included for comparison as well (simply denoted by "ScatNet"). Since the KTH-TIPS dataset is enhanced from CUReT, we use another dataset UIUC instead of CUReT for the comparison, and set different sample sizes (number of samples per class) to show how this factor influence the performances. The results presented in Table 6 demonstrate that DCR consistently performs better than that of ScatNet and other methods which are most competitive in former experiments, especially in the small sample size cases.

In general, there are two important conclusions which could be easily derived from the details of the results.

1. *DCR learns a good dictionary for collaborative representation.* In all the experiments, dictionary learning in DCR consistently and greatly improves the performance of collaborative representation (compared with CRC).
2. *DCR appears less over-fitting and more effective than other dictionary learning models in our experiments*, which is very significant, especially on those datasets with few samples per class (such as AR, Food, KTH-TIPS_5, UIUC_5, and UIUC_10) and large within-class variations (such as iLIDS-AA, KTH-TIPS, CUReT and Food).

Table 7. Computational cost comparison with all the related methods on all concerned classification tasks. The best results are in bold, and the best results for dictionary learning based methods are underlined.

	Training Time (ms/sample)						Test Time (ms/sample)					
Dataset	SRC	CRC	FDDL	LC-KSVD	DL-COPAR	DCR	SRC	CRC	FDDL	LC-KSVD	DL-COPAR	DCR
Extended Yale B	**0**	**0**	2386	116.6	1274	57.6	3093	16.9	1257	**0.56**	18.9	9.7
iLIDS-AA	**0**	**0**	134715	6059	539.4	124.2	8420	**9.8**	10349	21.1	48.7	16.6
KTH-TIPS	**0**	**0**	1138	265.2	3154	25.0	2830	**0.18**	2371	4.4	6.9	8.8
Swedish Leaf	**0**	**0**	1753	219.6	1806	346.1	4149	**0.77**	3470	3.2	9.2	7.8

4.4 Computational cost

We choose a representative dataset for each problem to compare the actual training/test time for all the adopted sparse/collaborative representation based models. The results are averaged over the 10 trials if applicable, and we report them in the "per sample" manner to eliminate the influence of dataset size. All the methods compared are implemented in Matlab and ran on a 2.67 GHz machine with 20GB memory (more than actually needed). The results listed in Table 7 show that the learning model of DCR is generally more efficient than those of other dictionary learning methods (especially its analogues FDDL and DL-COPAR). Though LC-KSVD is very fast in the testing stage as it needs only a linear projection, the test time of DCR is comparable to that of LC-KSVD. This is unlike FDDL which needs expensive optimization even at the test stage. We can verify the correctness of the theoretical complexity (shown in Table 1) by comparing it with the actual computational time. Take "Extended Yale B" as an example, the theoretical complexity for training is $\mathcal{O}(2.72 \times 10^{10})$ while the actual training time is about 6.8 times of 2.72×10^{10}, showing that they match each other very well.

5 Conclusions

We have proposed a novel dictionary learning model DCR for classification, which to the best of our knowledge is the first one for the l_2-norm based collaborative representation. Extensive experimental results on 9 benchmark datasets for 4 types of tasks have shown that DCR is more effective and less over-fitting than the state-of-the-art. Its performance is also superior to the latest results from some unrelated methods. Moreover, DCR learns its dictionary faster than the other related dictionary learning models due to the closed-form solutions for each sub-problem in the alternative optimization. Future work includes a comparison of the concerned models on how their performances change when the trade-off parameters ar e tuned, which may reveal new interesting findings on the effectiveness of each component and the models' sensitivity to these parameters.

Acknowledgement. This work was supported by "R&D Program for Implementation of Anti-Crime and Anti-Terrorism Technologies for a Safe and Secure Society", Funds for integrated promotion of social system reform and research and development of the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.

References

1. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE TPAMI **31** (2009) 210–227
2. Zhang, L., Yang, M., Feng, X.: Sparse representation or collaborative representation: which helps face recognition? In: ICCV. (2011)
3. Wright, J., Ma, Y., Mairal, J., Spairo, G., Huang, T., Yan, S.: Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE (2010)
4. Jiang, Z., Lin, Z., Davis, L.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: CVPR. (2011) 1697 –1704
5. Yang, M., Zhang, L., Feng, X., Zhang, D.: Fisher discrimination dictionary learning for sparse representation. In: ICCV. (2011) 543 –550
6. Kong, S., Wang, D.: A dictionary learning approach for classification: Separating the particularity and the commonality. In: ECCV. (2012)
7. Yang, M., Zhang, L., Yang, J., Zhang, D.: Metaface learning for sparse representation based face recognition. In: ICIP. (2010) 1601 –1604
8. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: CVPR. (2010) 3501 –3508
9. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. In: NIPS. (2009) 1033–1040
10. Zhang, Q., Li, B.: Discriminative k-svd for dictionary learning in face recognition. In: CVPR. (2010) 2691 –2698
11. Szlam, A., Sapiro, G.: Discriminative k-metrics. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09, New York, NY, USA, ACM (2009) 1009–1016
12. Yu, G., Sapiro, G., Mallat, S.: Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. Image Processing, IEEE Transactions on **21** (2012) 2481–2499
13. Bezdek, J.C., Hathaway, R.J.: Convergence of alternating optimization. Neural, Parallel Sci. Comput. **11** (2003) 351–368
14. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE TPAMI **23** (2001) 643 –660
15. Martinez, A., Benavente, R.: The ar face database. CVC Technical Report 24 (1998)
16. Bak, S., Corvee, E., Bremond, F., Thonnat, M.: Boosted human re-identification using riemannian manifolds. Image and Vision Computing **30** (2012) 443 – 452
17. Wu, Y., Minoh, M., Mukunoki, M., Lao, S.: Robust object recognition via third-party collaborative representation. In: ICPR. (2012)
18. Wu, Y., Minoh, M., Mukunoki, M., Li, W., Lao, S.: Collaborative sparse approximation for multiple-shot across-camera person re-identification. In: AVSS. (2012) 209 –214
19. Qi, X., Xiao, R., Guo, J., Zhang, L.: Pairwise rotation invariant co-occurrence local binary pattern. In: ECCV. (2012)
20. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV **73** (2007) 213–238
21. Caputo, B., Hayman, E., Fritz, M., Eklundh, J.O.: Classifying materials in the real world. Image and Vision Computing **28** (2010) 150 – 163
22. Wu, J., Rehg, J.: Centrist: A visual descriptor for scene categorization. IEEE TPAMI **33** (2011) 1489–1501
23. Yang, S., Chen, M., Pomerleau, D., Sukthankar, R.: Food recognition using statistics of pairwise local features. In: CVPR. (2010) 2249–2256
24. Bruna, J., Mallat, S.: Invariant scattering convolution networks. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 1872–1886