

Relaxed Collaborative Representation for Pattern Classification

Meng Yang, Lei Zhang, David Zhang, Shenlong Wang

Depart. of Computing, The Hong Kong Polytechnic University, Hong Kong

{csmyang, cslzhang}@comp.polyu.edu.hk

Abstract

Regularized linear representation learning has led to interesting results in image classification, while how the object should be represented is a critical issue to be investigated. Considering the fact that the different features in a sample should contribute differently to the pattern representation and classification, in this paper we present a novel relaxed collaborative representation (RCR) model to effectively exploit the similarity and distinctiveness of features. In RCR, each feature vector is coded on its associated dictionary to allow flexibility of feature coding, while the variance of coding vectors is minimized to address the similarity among features. In addition, the distinctiveness of different features is exploited by weighting its distance to other features in the coding domain. The proposed RCR is simple, while our extensive experimental results on benchmark image databases (e.g., various face and flower databases) show that it is very competitive with state-of-the-art image classification methods.

1. Introduction

Inspired by the sparse coding (or sparse representation) mechanism of human vision system [18] [23], and with the rapid development of l_1 -norm minimization techniques in recent years, the sparse coding methods have been successfully used in various image restoration applications [13]. Many efforts have also been made to apply sparse coding methods to pattern classification tasks, such as signal classification [7], face recognition (FR) [25] and image classification [26], etc. Though interesting classification results have been achieved, more investigations need to be made in order for a clearer understanding about the relationship between object representation and classification.

In the application of FR, which is one of the most active research topics in computer vision and pattern recognition [30], sparse representation based algorithms [25][27] have achieved much superior performance (i.e., robustness to illumination changes, random pixel corruption, block occlusion and real disguise, etc.) to representative FR methods,

such as Eigenface [1], Nearest Subspace [10] and SVM [6], etc. In the pioneer work of sparse representation based classification (SRC) [25], the query face image is coded as a sparse linear combination of all the training samples via l_1 -norm minimization; particularly, in SRC an identity matrix can be introduced to code the outlier pixels, making SRC robust to face occlusion and corruption. The success of SRC boosts the research of sparsity based pattern classification, and many works have been consequently reported, for examples, l_1 -graph for clustering and subspace learning [3], sparse image classification [26], and robust sparse coding for FR [27], etc.

Despite the wide use of sparse representation for classification, recently researchers have begun to question the role of sparsity in classification [29] [20]. In [29], it has been shown that it is the collaborative representation (i.e., representing the query image collaboratively by samples from all the classes) but not the l_1 -norm sparse representation that makes SRC effective for pattern classification. Using the non-sparse l_2 -norm to regularize the representation coefficients could lead to similar FR results to l_1 -norm regularization but this can significantly speed up the algorithm. The robustness to outliers (e.g., occlusion and corruption) in query face image actually comes from the sparsity constraint on coding residuals but not on the coding coefficients.

Without considering the robustness to outlier pixels in face images, both SRC [25] and collaborative representation based classification (CRC) [29] can be regarded as the regularized linear regression problem:

$$\min_{\alpha} \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_{l_p} \quad (1)$$

where $\mathbf{y} = [y_1; y_2; \dots; y_n]$ is the query image vector or its feature vector, \mathbf{D} is the dictionary whose columns are the training image vectors or their feature vectors, and λ is a scalar constant. When $p = 1$ or $p = 2$, Eq. (1) becomes the coding model of SRC or CRC, respectively. If we write \mathbf{D} as $\mathbf{D} = [\mathbf{r}_1; \mathbf{r}_2; \dots; \mathbf{r}_n]$, where \mathbf{r}_i is the i^{th} row of dictionary \mathbf{D} , then Eq. (1) could be rewritten as

$$\min_{\alpha} \sum_{i=1}^n (y_i - \mathbf{r}_i\alpha) + \lambda \|\alpha\|_{l_p} \quad (2)$$

From Eq. (2) we can clearly see that in SRC and CRC, all the feature elements y_i are enforced to share the same coding vector α over their associated sub-dictionaries (i.e., r_i). However, this requirement is too strong and it ignores the fact that the feature elements in a pattern share similarities but also have differences. Therefore, the feature elements should have similar coding coefficients so that they can jointly represent the same pattern, while their coding coefficients should have some diversity to reflect the distinctive property of different features (e.g., pixels in different spatial locations, different frequency features, the Gabor features along different orientations or scales, etc.). For instance, one can imagine that the occluded part of a face image should have very different coding coefficients compared to those of the non-occluded facial parts.

How object representation should be learned is a very important issue to pattern classification tasks. In some recent works which employ multiple types of features (say K types) for joint sparse representation and recognition [12][17][28], the mixed-norm regularization is adopted to optimize the coding coefficients. Two widely used mixed-norm regularizations are the $l_{1,2}$ -norm $\sum_i \|\alpha^i\|_2$ and the $l_{1,\infty}$ -norm $\sum_i \|\alpha^i\|_\infty$, where α^i is the i^{th} row of the coding coefficient matrix $\mathbf{A} = [\alpha_1, \alpha_2, \dots, \alpha_K]$ with α_j being the coding vector of the j^{th} feature vector \mathbf{y}_j over its associated dictionary. Although the $l_{1,2}$ -norm and $l_{1,\infty}$ -norm regularizations do not require the coding vectors α_j to be the same but require that the coding vectors are similar, they assume that every feature has the same contribution to coding, as well as to classification. Intuitively, the distinctiveness of different features should be considered in the coding process for a more robust recognition. The coding coefficients will be more discriminative if the different importance of the features can be exploited in the coding phase, which will consequently benefit the final classification accuracy.

In this paper, we propose a relaxed collaborative representation (RCR) model, which considers both the similarity and distinctiveness of different features in coding and classification stages. In the coding stage, apart from requiring that each feature can be well represented by its associated dictionary, which allows the diversity of coding vectors for different features, a weighted regularization term is introduced to enforce that the coding vectors from different features have a small variance, which accounts for the similarity between features. Meanwhile, the weights are optimized simultaneously with the coding to address the distinctiveness of different features. In the classification stage, we assign the query sample to the class which yields the lowest weighted coding residual. In addition, if the weights can be learned offline, then a closed-form solution of RCR could be obtained. Our extensive experiments on face recognition and object categorization show that the proposed RCR scheme has very competitive performance with state-of-the-

arts while having a low time complexity.

The rest of this paper is organized as follows. Section 2 briefly reviews some related works. Section 3 presents the model of RCR and its optimization algorithm. Section 4 makes some discussions. Section 5 performs experiments, and Section 6 concludes the paper.

2. Brief review of related works

The sparse representation based classification (SRC) method was presented in [25] for robust face recognition (FR). Denote by $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_c]$ the matrix formed by original training samples, where \mathbf{A}_i is the sub-set of training samples from class i , and c is the number of classes. Let \mathbf{y} be a query sample to be classified. In SRC, first \mathbf{y} is sparsely coded on \mathbf{A} via l_1 -minimization

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{y} - \mathbf{A}\alpha\|_2 + \lambda \|\alpha\|_1 \quad (3)$$

where λ is a scalar constant. Then classification is made by

$$\text{identity}(\mathbf{y}) = \arg \min_i \{e_i\} \quad (4)$$

where $e_i = \|\mathbf{y} - \mathbf{A}_i \hat{\alpha}_i\|_2$, $\hat{\alpha} = [\hat{\alpha}_1; \hat{\alpha}_2; \dots; \hat{\alpha}_c]$ and $\hat{\alpha}_i$ is the coefficient vector associated with class i .

Though it was claimed in [25] that the l_1 -norm sparsity imposed on coding coefficient α is the key for the success of SRC, recently it has been shown in [29] that it is the collaborative representation based classification (CRC), but not the l_1 -norm sparsity on α , that truly makes SRC effective for face classification. Using l_2 -norm to regularize α leads to similar FR results. The robustness to outliers in SRC actually comes from using l_1 -norm to model the coding residual, i.e., $\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{y} - \mathbf{A}\alpha\|_1 + \lambda \|\alpha\|_1$. Without considering the robustness to outliers, the coding model of CRC is

$$\hat{\alpha} = \arg \min_{\alpha} \|\mathbf{y} - \mathbf{A}\alpha\|_2 + \lambda \|\alpha\|_2^2 \quad (5)$$

The classification of CRC is performed by checking which class yields the minimal regularized reconstruction error, which is similar to that of SRC.

With multiple types of input features, a multi-task joint sparse representation based classification (MTJSRC) method was proposed in [28]. Suppose each sample has K different modalities of features. Denote by \mathbf{y}^k the k^{th} modality of feature vector to be coded, by \mathbf{A}^k the dictionary of the k^{th} modality of feature and by $\alpha^k = [\alpha_1^k; \alpha_2^k; \dots; \alpha_c^k]$ the coding vector of \mathbf{y}^k over \mathbf{A}^k , where α_j^k is the sub-vector associated with class j . Let $\alpha_j = [\alpha_j^1, \alpha_j^2, \dots, \alpha_j^K]$. The MTJSRC with $l_{1,2}$ -norm regularization is formulated as [28]

$$\min_{\alpha^k} \sum_{k=1}^K \left\| \mathbf{y}^k - \mathbf{A}^k \alpha^k \right\|_2^2 + \lambda \sum_{j=1}^c \|\alpha_j\|_2 \quad (6)$$

which expects that α_j^k for different modalities k are similar, and those α_j for different classes j are sparse. The classification of MTJSRC is performed by checking which class yields the minimal overall reconstruction error of K modalities.

3. Relaxed collaborative representation

3.1. Relaxed collaborative representation model

It is reasonable to assume that the different features extracted from one sample, as well as their corresponding sub-dictionaries extracted from the whole dictionary, may share some similarity. Therefore one can assume that the representation coefficients of those features over their associated sub-dictionaries should be similar. This can make the representation stable. On the other hand, those different features (e.g., features of different spatial locations, frequency bands, orientations, scales and modalities) can be very distinctive from each other, so that we should allow their representations over the associated dictionaries have enough diversity. This can make the representation flexible. Overall, a good balance between *stability* and *flexibility* will lead to a stable and accurate representation for accurate recognition tasks.

To achieve the above goal and exploit the distinctiveness of different features in linear regression, we propose the following term to regularize the coding vectors of different features over their associated dictionaries:

$$\min_{\alpha_k} \sum_{k=1}^K \omega_k \|\alpha_k - \bar{\alpha}\|_2^2 \quad (7)$$

where $\alpha_k, k = 1, 2, \dots, K$ is the coding vector of the k^{th} feature vector \mathbf{y}_k over the k^{th} dictionary \mathbf{D}_k , $\bar{\alpha}$ is the mean vector of all α_k , and ω_k is the weight assigned to the k^{th} feature. It is easy to see that Eq. (7) aims to reduce the variance of coding vectors α_k , making them similar to each other; at the same time, the weight ω_k is used to indicate the distinctiveness of feature \mathbf{y}_k . It is easy to see that group coding [2] or joint coding [12][17][28] can only reduce the variance of coding vectors α_k but ignore these features' distinctiveness, which is very important for classification (e.g., FR with disguise). Intuitively, if \mathbf{y}_k is more similar to other features, ω_k should be bigger to enforce α_k closer to the mean $\bar{\alpha}$; if \mathbf{y}_k is less similar to other features, ω_k should be smaller so that α_k can vary more from others.

With the regularizer in Eq. (7), the proposed relaxed collaborative representation (RCR) is

$$\min_{\alpha_k, \omega_k} \sum_{k=1}^K \left(\|\mathbf{y}_k - \mathbf{D}_k \alpha_k\|_2^2 + \lambda \|\alpha_k\|_2^2 \right) + \tau \sum_{k=1}^K \omega_k \|\alpha_k - \bar{\alpha}\|_2^2 \quad \text{s. t. } \text{prior} \{\omega_k\} \quad (8)$$

where λ and τ are positive constants and *prior* $\{\omega_k\}$ means the prior made on weights ω_k . Note that in Eq. (8) we use

l_2 -norm to regularize α_k since it has been shown in [29] that the l_1 -norm sparsity on α_k is not necessary but makes the coding complexity high.

According to prior we impose on ω_k , RCR has three special cases.

1) *RCR with strong prior*. In this case, the weights ω_k can be pre-learned by using a validation dataset, and then Eq. (8) can have a closed-form solution since all the terms are of l_2 -norm.

2) *RCR with moderate prior*. In this case, the values of these weights are unknown beforehand. However, some prior information of these weights (e.g., $\omega_i > \omega_j$ for some $i \neq j$) could be known empirically. To reduce the risk of overweighting some features while ignoring other features, we regularize the weights based on the maximum entropy principle (here we assume that the weight ω_k is normalized in $[0, 1]$). Let $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_K]^T$. We set the prior of weights as

$$\begin{cases} -\sum_{k=1}^K \omega_k \ln \omega_k > \sigma \\ \mathbf{u}_l \leq \boldsymbol{\Phi} \boldsymbol{\omega} \leq \mathbf{u}_c \\ 0 \leq \boldsymbol{\omega} \end{cases} \quad (9)$$

where the matrix $\boldsymbol{\Phi}$ reflects the relative relation between ω_k . For instance, if $\omega_2 \geq \omega_1 \geq 0$ when there are two feature vectors, then $\boldsymbol{\Phi} = [1, -1]$, $u_c = 0$, $u_l = -\infty$.

3) *RCR with weak prior*. No prior information about the weights is known except that we regularize their entropy:

$$-\sum_{k=1}^K \omega_k \ln \omega_k > \sigma \quad (10)$$

3.2. Optimization algorithm

In the case of RCR with strong prior, since the weights $\boldsymbol{\omega}$ are pre-learned, we only need to solve α_k and a global optimum can be reached, as given in Eq. (12). For the other two cases, the objective function in Eq. (8) can be solved by alternatively optimizing $\boldsymbol{\omega}$ and α_k , i.e., updating the coding vector α_k by fixing the weights $\boldsymbol{\omega}$, and updating the weights $\boldsymbol{\omega}$ by fixing α_k . Such a process is iterated until the solutions of $\boldsymbol{\omega}$ and α_k converge to some local minimum.

First, if the weights $\boldsymbol{\omega}$ are known, the optimization of Eq. (8) becomes

$$\min_{\alpha_k} \sum_{k=1}^K \|\mathbf{y}_k - \mathbf{D}_k \alpha_k\|_2^2 + \lambda \|\alpha_k\|_2^2 + \tau \omega_k \|\alpha_k - \bar{\alpha}\|_2^2 \quad (11)$$

from which a closed-form solution for $k = 1, 2, \dots, K$ could be derived:

$$\alpha_k = \alpha_{0,k} + \tau \frac{\omega_k}{\sum_{\eta=1}^K \omega_{\eta}} \mathbf{P}_k \mathbf{Q} \sum_{\eta=1}^K \omega_{\eta} \alpha_{0,\eta} \quad (12)$$

$$\bar{\alpha} = \sum_{k=1}^K \omega_k \alpha_k / \sum_{k=1}^K \omega_k \quad (13)$$

where $P_k = (D_k^T D_k + I(\lambda + \tau\omega_k))^{-1}$, $\alpha_{0,k} = P_k D_k^T \mathbf{y}_k$, $Q = (I - \sum_{\eta=1}^K \varpi_\eta P_\eta)^{-1}$, $\varpi_\eta = \frac{\tau\omega_\eta^2}{\sum_{k=1}^K \omega_k}$. For a detailed derivation, please refer to **Appendix A**.

Once the coding vectors α_k are obtained by Eq. (12), the coding weights ω can then be updated. For RCR with moderate prior, the objective function in Eq. (8) is reduced to:

$$\min_{\omega_k} \sum_{k=1}^K \tau\omega_k \|\alpha_k - \bar{\alpha}\|_2^2 + \gamma \sum_{k=1}^K \omega_k \ln \omega_k \quad (14)$$

s. t. $\mathbf{u}_l \leq \Phi\omega \leq \mathbf{u}_c; \mathbf{0} \leq \omega$

which could be solved effectively by the toolbox MOSEK (www.mosek.com). Here $\gamma > 0$ is the Lagrange multiplier. For RCR with weak prior, the objective function in Eq. (8) is reduced to

$$\min_{\omega_k} \sum_{k=1}^K \tau\omega_k \|\alpha_k - \bar{\alpha}\|_2^2 + \gamma\omega_k \ln \omega_k \quad (15)$$

and the weights could be directly updated as

$$\omega_k = \exp \left\{ -1 - \tau \|\alpha_k - \bar{\alpha}\|_2^2 / \gamma \right\} \quad (16)$$

The algorithm of RCR optimization is summarized in Algorithm 1. The RCR solution converges since the two alternative optimizations in it are both convex.

Algorithm 1 Algorithm of Relaxed Collaborative Representation (RCR)

- 1: **Input:** Dictionary D_k and feature vectors \mathbf{y}_k of the query sample, $k = 1, 2, \dots, K$. An initialization of the weight vector $\omega^{(0)}$.
 - 2: **While** not converged **do**
 - 3: updating coding vectors Eq. (12);
 - 4: updating weights via Eq.(14) or (16);
 - 5: checking convergence condition:
 $\|\omega^{(t+1)} - \omega^{(t)}\|_2 / \|\omega^{(t)}\|_2 < \delta_\omega$
 where $\omega^{(t)}$ is the weight vector in the t^{th} iteration.
 - 6: **end while**
 - 7: **Output:** $\alpha_k, k = 1, 2, \dots, K$ and ω .
-

3.3. Classification

The classification of RCR is based on the overall coding error for each class. For the query sample \mathbf{y} , the overall coding error by class i is computed as

$$e_i = \sum_{k=1}^K \omega_k \|\mathbf{y}_k - D_k^i \alpha_k^i\|_2^2 \quad (17)$$

where D_k^i is the sub-set of the dictionary D_k associated with class i , and α_k^i the coefficient vector α_k associated with class i . So the classification is done via

$$\text{identity}(\mathbf{y}) = \arg \min_i \{e_i\} \quad (18)$$

4. Discussion of RCR

4.1. Feature

To obtain the feature vectors \mathbf{y}_k of the query sample \mathbf{y} , one straightforward way is to divide the image into multiple blocks, and stretch each block to a vector \mathbf{y}_k for RCR. This multi-block RCR simply uses the intensity in each block as the feature to identify the identity of the query image. In [25] [29] and [8], the face image is partitioned into 8 blocks to do FR with disguise. However, these methods process the 8 blocks independently without considering their relationships. Better performance could be achieved by multi-block RCR because it considers the relationship between the different blocks in coding and classification.

Another way to obtain \mathbf{y}_k is applying different feature extractors to \mathbf{y} , and the output of each feature extractor is taken as one \mathbf{y}_k . Compared with multi-block RCR, this multi-feature RCR can be used to do more challenging tasks, for examples, large-scale and real-world FR and object category, where using only a single type of features often cannot accomplish the task satisfyingly. In MTJSRC [28], multiple features are employed for joint sparse representation via $l_{1,2}$ -norm, but it does not consider the distinctiveness of different features in the coding. Multi-feature RCR could overcome this limitation by balancing automatically the similarity and distinctiveness of different features.

4.2. Weight

In RCR, the weights ω could be learned online, or learned offline via a validation set. Multi-block RCR could be categorized as the case of RCR with weak prior for that it is hard to learn the weight for each spatial region. The weight should be adaptively determined when dealing with a query sample. Such an online weight learning in multi-block RCR is effective to handle the occlusions in FR. Multi-feature RCR could be categorized as the case of RCR with strong or moderate prior for that it is possible to know the effectiveness of a specific feature to a certain task. With a validation dataset, the weight of multi-feature RCR could be learned by the algorithm of RCR with moderate prior. Then the weights used in the testing set could be simply set as

$$\omega_k = \sum_{i=1}^{n_v} \omega_k^i / n_v \quad k = 1, 2, \dots, K \quad (19)$$

where ω_k^i is the weight of the k^{th} feature vector for the i^{th} validation sample, and n_v is the total number of validation samples.

4.3. Analysis of complexity

When the weight values of different features are known or pre-learned, RCR will have a closed-form solution of coding vector α_k as shown in Eq. (12), where the projection

matrices could be computed offline. Suppose that the size of D_k is $n_k \times m_k$, the time complexity of coding is only $O(\sum_{k=1}^K (3m_k^2 + m_k n_k))$, where computing all $P_k D_k^T y_k$ has complexity $O(\sum_{k=1}^K (m_k^2 + m_k n_k))$ and computing all $P_k Q \sum_{\eta=1}^K \omega_\eta \alpha_{0,\eta}$ has complexity $O(\sum_{k=1}^K 2m_k^2)$.

When the weights need to be updated online, the time complexity of RCR increases due to the iterative optimization, which involves the operation of matrix inverse. Fortunately, we find that it affects little the performance of RCR if P_k and Q are predefined in the iteration. In that case, the complexity of RCR with learning weight online is only $O(q \sum_{k=1}^K (3m_k^2 + m_k n_k))$, where q is the iteration number.

The experimental running speed of RCR is also very fast. For instance, its running time on the AR database (refer to Section 5.1) is 0.015 second (with known weights), 0.05 second (updating weights online with predefined P_k and Q), or 0.9 second (updating weights and P_k and Q online), respectively.

5. Experiments

To evaluate the effectiveness of our proposed RCR, we apply it to FR in controlled/uncontrolled environments and multi-class object recognition. For FR in controlled environment, multi-block RCR is employed. There are three parameters in multi-block RCR: λ , τ and γ (the Lagrange multiplier of the entropy constraint). In our experiments, λ and τ are set as 0.0005 and 0.005, respectively; γ is set as 0.1 for FR without occlusion and 0.001 for FR with disguise. For more challenging tasks such as FR in uncontrolled environment and object categorization, multi-feature RCR is employed, and the parameters, i.e., λ , τ and the weights ω , are learned from the validation set. The source code of this paper can be downloaded at www4.comp.polyu.edu.hk/~cslzhang/code.htm.

5.1. FR in controlled environment

In this section, we perform FR without and with occlusion on two benchmark face datasets captured in controlled environments: the Extended Yale B [5][9] and a subset of AR [14]. The Extended Yale B database contains about 2414 frontal images (cropped to 54×48) of 38 individuals; and the subset of AR contains two-session data of 50 male and 50 female subjects (each person has 26 pictures with the normalized size as 60×43).

a) *FR without occlusion*: The SVM (linear kernel) is used as the baseline, and the methods of SRC [25], CRC [29], MTJSRC [28] and LRC [8] are used to compare with the proposed RCR. The multi-block RCR is used here, and we simply divide the face image into 1×4 blocks.

For each subject of Extended Yale B, we randomly selected N_{tr} images for training with the remaining images

for testing. The recognition rates of different methods versus training number are shown in Table 1. RCR and MTJSRC [28] have the highest recognition rates in all cases. Consistent with [29], almost the same recognition rates are achieved by SRC and CRC, both of which are better than LRC. The best recognition rates of SRC, CRC, LRC, SVM, MTJSRC and RCR are 92.0%, 92.4%, 89.0%, 88.1% 93.6% and 93.6%, respectively. In addition, the average variance of coding coefficients for all testing samples, i.e., $\frac{1}{4} \sum_{k=1}^4 \|\alpha_k - \bar{\alpha}\|_2^2$, of RCR and MTJSRC are 0.534 and 0.540, respectively, when $N_{tr} = 25$, and are 0.692 and 0.710, respectively, when $N_{tr} = 20$, which shows that RCR achieves higher stability.

For each subject of AR, the images only with illumination and expression changes are selected for experiments. The samples from Session 1 are used for training and that from Session 2 for testing. The recognition rates of all the competing methods are listed in Table 2. It can be seen that RCR outperforms all the other methods (except for MTJSRC) by more than 2%. Similar to the results in Extended Yale B, the performance of RCR and MTJSRC is similar. The reason may be that each block in face images has similar weight (or contribution) to FR without occlusion.

b) *FR with real face disguise*: As in [25][29], 800 images (about 8 samples per subject) with only expression changes selected from the subset of AR are used for training, while two separate subsets (with sunglasses or scarf) of 200 images (1 sample per subject per Session, with neutral expression, as shown in Fig.1(a)) for testing. Here the images were resized to 83×64 , and partitioned into 4×2 blocks (refer to Fig.1(b)) as [25] for all the competing methods (i.e., MTJSRC, RCR, the block versions of SRC, CRC and LRC). The recognition rates of these five methods are listed in Table 3. RCR gets the best performance, with 2%, 4%, 2.5%, and 12% average improvement over SRC, CRC, LRC and MTJSRC, respectively. Compared to SRC, CRC, and LRC, which separately represent each block and fuse all blocks' results via voting or minimal reconstruction error, RCR could jointly represent the blocks for more discriminative coding. Compared to MTJSRC, which treats each block equally and uses the mixed $l_{1,2}$ -norm to enforce the similarity between features, RCR automatically learns weights to distinguish the importance between occluded and non-occluded blocks and minimizes the variance of coding vectors to enforce similarity. The results clearly show that RCR is much more robust than MTJSRC in FR with occlusion.

5.2. FR in uncontrolled environment

In this section, we evaluate the performance of multi-feature RCR in large-scale and real-world face databases: FRGC 2.0 [19] and LFW-a [24]. Four features, i.e., intensity value, low-frequency Fourier feature [21], Gabor mag-

N_{tr}	10	15	20	25
SVM	60.0%	67.1%	76.5%	88.1%
SRC	84.6%	84.2%	91.3%	92.0%
CRC	84.8%	84.7%	91.2%	92.4%
LRC	82.4%	81.8%	87.0%	89.0%
MTJSRC	87.3%	87.4%	91.5%	93.6%
RCR	86.8%	87.2%	92.3%	93.6%

Table 1. Face recognition rates on the Extended Yale B database.

SVM	SRC	CRC	LRC	MTJSRC	RCR
87.1%	93.7%	93.3%	76.4%	95.8%	95.9%

Table 2. Face recognition rates on the AR database.



Figure 1. (a) The testing samples with sunglasses and scarves in the AR database; (b) partitioned testing samples.

Method	Sunglasses	Scarf
SRC	97.5%	93.5%
CRC	91.5%	95%
LRC	95.5%	94.5%
MTJSRC	80.5%	90.5%
RCR	98.5%	96.5%

Table 3. Recognition rates by competing methods on the AR database with disguise.



Figure 2. Samples of FRGC 2.0 and LFW. (a) and (b) are samples in target and query sets of FRGC 2.0; (c) and (d) are samples in training and testing sets of LFW.

nitude [11] and LBP [22], are used for all the competing methods, SRC, CRC, LRC, MTJSRC and RCR. For each feature, we adopt a “divide and conquer” strategy: first extract the discrimination-enhanced feature in each block (each image is partitioned into 2×2 blocks) via LDA [1], and then concatenate all blocks’ features as the final feature vector.

a) FRGC: FRGC version 2.0 [19] is a large-scale face database designed with uncontrolled indoor and outdoor setting. We use a subset (352 subjects having no less than 15 samples in original target set) of Experiment 4, which is the most challenging data set in FRGC 2.0 with large lighting variations, aging and image blur, as shown in Figs. 2(a)

N_{ta}	SRC	CRC	LRC	MTJSRC	RCR
15	94.9%	94.4%	95.1%	94.3%	95.3%
10	88.0%	87.4%	87.3%	87.7%	88.4%
5	83.3%	82.9%	82.9%	84.7%	85.3%

Table 4. Face recognition rates on FRGC2.0 Exp 4.

and 2(b). The selected target set contains 5280 samples, and the query set has 7606 samples. We use half of original validation set to learn projection of LDA [1] and the weight values of RCR. For MTJSRC, we also learn the weights to weight the coding error for better classification.

Three tests with the first N_{ta} (e.g., 5, 10, and 15) target samples per subject are performed. The recognition rates of SRC, CRC, LRC, MTJSRC and RCR using the combination of four features are listed in Table 4. RCR outperforms all the other methods although the improvement is not so large since there are no occlusion, misalignment and pose variations in the query set. It can also be seen that when the number of target samples is high enough (i.e., 15 samples per subject), all the methods could achieve good performance (more than 94% recognition accuracy).

b) LFW: Labeled Faces in the Wild (LFW) is a large-scale database of face photographs designed for unconstrained FR with variations of pose, illumination, expression, misalignment and occlusion, etc (shown in Figs. 2(c) and 2(d)). Two subsets of aligned LFW [24] are used in the experiments. In subset 1 which consists of 311 subjects with no less than 6 samples per subject, we use the first 5 samples as training data and the remaining samples as testing data. In subset 2 which consists of 143 subjects with no less than 11 samples per subject, we use the first 10 samples as training data and the remaining samples as testing data.

We use the learned weights in FRGC dataset for RCR and MTJSRC here. Table 5 lists the results by different methods on the two subsets. RCR has at least 6% and 7% improvements over SRC, CRC, and LRC in subset 1 and subset 2, respectively, which demonstrates that it is not effective to let different features of a sample share the same representation coefficients. Compared with MTJSRC, which allows different features to have different but similar representation coefficients, and uses the weighted coding error to do classification, RCR has more than 6% (3%) higher rate than it in subset 1 (subset 2). This validates that the proposed RCR model can more effectively exploit the similarity and distinctiveness of different features for coding and classification.

5.3. Object categorization

At last, let’s validate the effectiveness of the proposed method on multi-class object categorization. The two Oxford flower datasets [15][16] are used

	SRC	CRC	LRC	MTJSRC	RCR
subset 1	53.0%	54.5%	48.7%	54.8%	61.0%
subset 2	72.2%	73.0%	60.5%	77.4%	80.6%

Table 5. Face recognition rates on LFW.

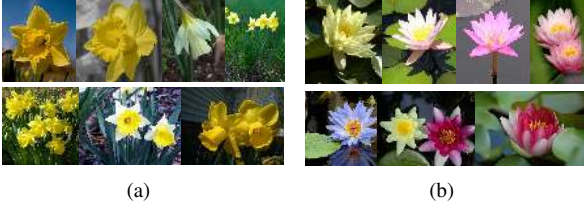


Figure 3. Samples from Oxford flower data sets. (a) Some samples of 'daffodil' in 17 category; and (b) some samples of 'water lily' in 102 category.

here, some samples of which are show in Figs.3(a) and 3(b). For these two data sets, we adopt the default experimental settings provided on the website (www.robots.ox.ac.uk/~vgg/data/flowers), including the training, validation, and test splits and the multiple features. It should be noted that these features are only extracted from flower regions which are well cropped by preprocessing of segmentation.

For a fair comparison with MTJSRC [28], we also extended RCR to its kernel versions for the experiments on these two datasets. In the case of direct kernel version, it is easy to see that the two terms in Eq.(12), $\mathbf{P}_k = (\mathbf{D}_k^T \mathbf{D}_k + \mathbf{I}(\lambda + \tau\omega_k))^{-1}$ and $\boldsymbol{\alpha}_{0,k} = \mathbf{P}_k \mathbf{D}_k^T \mathbf{y}_k$ could be transformed into $\mathbf{P}_k = (\mathbf{G}_k + \mathbf{I}(\lambda + \tau\omega_k))^{-1}$ and $\boldsymbol{\alpha}_{0,k} = \mathbf{P}_k \mathbf{h}_k$, where $\mathbf{G}_k = \phi_k(\mathbf{D}_k)^T \phi_k(\mathbf{D}_k)$, $\mathbf{h}_k = \phi_k(\mathbf{D}_k)^T \phi_k(\mathbf{y}_k)$, and ϕ_k is the kernel mapping function for the k^{th} modality of feature. Another kernel version of RCR is column generation, where we directly replace the k^{th} modality training data and testing data as their associated kernel matrices: $\mathbf{D}_k = \mathbf{G}_k$, and $\mathbf{y}_k = \mathbf{h}_k$. Here the kernel matrices are computed as $\exp(-\chi^2(x, x')/\mu)$, where μ is set to be the mean value of the pairwise χ^2 distances on the training set. We denote the direct kernel version of RCR as RCR-DK, and the column generation version of RCR as RCR-CG.

a) *17 category data set*: This set contains 17 species of flowers with 80 images per class. As in [28], we directly use the χ^2 distance matrices of seven features (i.e., HSV, HOG, SIFTint, SIFTbdy, color, shape and texture vocabularies) as inputs, and perform the experiments based on the three predefined training, validation, and test splits. The results (mean and variance) of RCR compared with other state-of-the-arts are presented in Table 6. We can see that both MTJSRC and RCR have much improvement over other methods (with about 2% improvement), while MTJSRC is slightly better than RCR.

Methods	Accuracy (%)
SRC Combination	85.9 \pm 2.2
MKL [4]	85.2 \pm 1.5
CG-Boost [4]	84.8 \pm 2.2
LPBoost [4]	85.4 \pm 2.4
MTJSRC-RKHS [28]	88.1 \pm 2.3 (86.8 \pm 1.8)
MTJSRC-CG [28]	88.9 \pm 2.9 (88.2 \pm 2.3)
RCR-DK	87.6 \pm 1.8 (87.4 \pm 1.3)
RCR-CG	88.0 \pm 1.6 (87.9 \pm 1.8)

Table 6. The categorization accuracy on the 17 category Oxford Flowers data set. The results in bracket are obtained under equal feature weights.

Methods	Accuracy (%)
SRC Combination	70.0
MKL [16]	72.8
MTJSRC-RKHS [28]	73.8 (71.5)
MTJSRC-CG [28]	74.1 (71.2)
RCR-DK	74.1 (71.1)
RCR-CG	75.0 (72.6)

Table 7. The categorization accuracy on the 102 category Oxford Flowers data set. The results in bracket are obtained under equal feature weights.

b) *102 category data set*: This set consists of 102 flower classes with 8198 images in total (40-250 images per class). As in [28], the χ^2 distance matrices of four features (i.e., HSV, HOG, SIFTint, and SIFTbdy) along with a predefined training, validation and test split, are directly used in the experiment. The comparison of RCR with other competing methods is shown in Table 7. It can be seen that RCR achieves the best performance, followed by MTJSRC. Specifically, RCR-CG has about 1% improvement over MTJSRC-CG. In addition, the learned feature weights are very beneficial to the final classification, 3% improvement for RCR-DK and 2.4% improvement for RCR-CG. The learned weights for RCR-DK (RCR-CG) are 0.2, 1.6, 1.5 and 0.9 (0.7, 1.6, 1.3 and 0.7), which show that the features of HOG and SIFTint are more discriminative.

6. Conclusion

In this paper, we proposed a relaxed collaborative representation model (RCR) for pattern classification, which effectively exploits the similarity and distinctiveness of different features for coding and classification. While allowing each feature vector to be flexibly coded over its associated dictionary, a novel regularization term was introduced to enforce the coding vectors having a small variance, and distinguish the distinctiveness of different features by adaptive weighting. Algorithms to optimize the proposed RCR were presented, and the experimental results on face recognition in controlled and uncontrolled environments and multi-class

object categorization clearly demonstrated the competitiveness of RCR to many state-of-the-art methods.

References

- [1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE PAMI*, 19(7):711–720, 1997.
- [2] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *NIPS*, 2009.
- [3] B. Cheng, J. C. Yang, S. C. Yan, Y. Fu, and T. Huang. Learning with l_1 -graph for image analysis. *IEEE IP*, 19(4):858–866, 2010.
- [4] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.
- [5] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE PAMI*, 23(6):643–660, 2001.
- [6] B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machine: Global versus component-based approach. In *ICCV*, 2001.
- [7] K. Huang and S. Aviyente. Sparse representation for signal classification. In *NIPS*, 2006.
- [8] N. Imran, T. Roberto, and B. Mohammed. Linear regression for face recognition. *IEEE PAMI*, 32(11):2106–2010, 2010.
- [9] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE PAMI*, 27(5):684–698, 2005.
- [10] S. Z. Li. Face recognition based on nearest linear combinations. In *CVPR*, 1998.
- [11] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE IP*, 11(4):467–476, 2002.
- [12] H. Liu, M. Palatucci, , and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *ICML*, 2009.
- [13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, 2009.
- [14] A. Martinez and R. benavente. The AR face database. Technical Report 24, CVC, 1998.
- [15] M. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006.
- [16] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICCVGIP*, 2008.
- [17] G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Journal of Statistics and Computing*, 20(2):231–252, 2010.
- [18] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997.
- [19] P. J. Phillips, P. J. Flynn, W. T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. J. Worek. Overview of the face recognition grand challenge. In *CVPR*, 2005.
- [20] R. Rigamonti, M. Brown, and V. Lepetit. Are sparse representations really relevant for image classification? In *CVPR*, 2011.
- [21] Y. Su, S. G. Shan, X. L. Chen, and W. Gao. Hierarchical ensemble of global and local classifiers for face recognition. *IEEE IP*, 18(8):1885–1896, 2009.
- [22] A. Timo, H. Abdenour, and P. Matti. Face recognition with local binary patterns. In *ECCV*, 2004.
- [23] W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *SCIENCE*, 287(5456):1273–1276, 2000.
- [24] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2009.
- [25] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE PAMI*, 31(2):210–227, 2009.
- [26] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.
- [27] M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *CVPR*, 2011.
- [28] X. T. Yuan and S. C. Yan. Visual classification with multi-task joint sparse representation. In *CVPR*, 2010.
- [29] L. Zhang, M. Yang, and X. C. Feng. Sparse representation or collaborative representation which helps face recognition? In *ICCV*, 2011.
- [30] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Survey*, 35(4):399–458, 2003.

Appendix A: The closed-form solution of Eq.(11):

Let $\mathbf{P}_k = (\mathbf{D}_k^T \mathbf{D}_k + \mathbf{I}(\lambda + \tau\omega^k))^{-1}$ and $\alpha_{0,k} = \mathbf{P}_k \mathbf{D}_k^T \mathbf{y}_k$, where \mathbf{I} is an identity matrix. By optimizing Eq.(11), we could get $\bar{\alpha} = \sum_{k=1}^K \omega_k \alpha_k / \sum_{k=1}^K \omega_k$ and $\alpha_k = \alpha_{0,k} + \tau\omega_k \mathbf{P}_k \sum_{k=1}^K \omega_k \alpha_k / \sum_{k=1}^K \omega_k$.

By summing $\omega_k \alpha_k$, $k = 1, 2, \dots, K-1$, we could get $\sum_{\eta=1}^{K-1} \omega_\eta \alpha_\eta = \sum_{\eta=1}^{K-1} \omega_\eta \alpha_{0,\eta} + \omega_K \sum_{\eta=1}^{K-1} \varpi_\eta \mathbf{P}_\eta \alpha_K$, $+ \sum_{\eta=1}^{K-1} \varpi_\eta \mathbf{P}_\eta \sum_{k=1}^{K-1} \omega_k \alpha_k$,

where $\varpi_\eta = \omega_\eta \tau \omega_\eta / \sum_{k=1}^K \omega_k$.

Then we have $(\mathbf{I} - \sum_{\eta=1}^{K-1} \varpi_\eta \mathbf{P}_\eta) \sum_{\eta=1}^{K-1} \omega_\eta \alpha_\eta = \sum_{\eta=1}^{K-1} \omega_\eta \alpha_{0,\eta} + \omega_K \sum_{\eta=1}^{K-1} \varpi_\eta \mathbf{P}_\eta \alpha_K$, with which we could put $\sum_{\eta=1}^{K-1} \omega_\eta \alpha_\eta$ in $\alpha_K = \alpha_{0,K} + \frac{\tau\omega_K}{\sum_{k=1}^K \omega_k} \mathbf{P}_K (\sum_{\eta=1}^{K-1} \omega_\eta \alpha_\eta + \omega_K \alpha_K)$.

After some derivations, we could get

$$\alpha_K = \alpha_{0,K} + \frac{\tau\omega_K}{\sum_{\eta=1}^K \omega_\eta} \mathbf{P}_K (\mathbf{I} - \sum_{\eta=1}^K \varpi_\eta \mathbf{P}_\eta)^{-1} \sum_{\eta=1}^K \omega_\eta \alpha_{0,\eta}$$

Similarly, all the representation coefficients are

$$\alpha_k = \alpha_{0,k} + \frac{\tau\omega_k}{\sum_{\eta=1}^K \omega_\eta} \mathbf{P}_k (\mathbf{I} - \sum_{\eta=1}^K \varpi_\eta \mathbf{P}_\eta)^{-1} \sum_{\eta=1}^K \omega_\eta \alpha_{0,\eta}$$

where $k = 1, 2, \dots, K$.