Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Locality-Constrained Low-Rank Coding for Image Classification

Ziheng Jiang†, Ping Guo†, Lihong Peng⋆
† School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
⋆ School of Computer Science, National University of Defense Technology, Changsha 410073, China
ziheng.j@gmail.com; pguo@ieee.org; penglihong@nudt.edu.cn

Abstract

Low-rank coding (LRC), originated from matrix decomposition, is recently introduced into image classification. Following the standard bag-of-words (BOW) pipeline, when coding the data matrix in the sense of low-rankness incorporates contextual information into the traditional BOW model, this can capture the dependency relationship among neighbor patches. It differs from the traditional sparse coding paradigms which encode patches independently. Current LRC-based methods use ℓ_1 norm to increase the discrimination and sparseness of the learned codes. However, such methods fail to consider the local manifold structure between data space and dictionary space. To solve this problem, we propose a locality-constrained low-rank coding (LCLR) algorithm for image representations. By using the geometric structure information as a regularization term, we can obtain more discriminative representations. In addition, we present a fast and stable online algorithm to solve the optimization problem. In the experiments, we evaluate LCLR with four benchmarks, including one face recognition dataset (extended Yale B), one handwritten digit recognition dataset (USPS), and two image datasets (Scene13 for scene recognition and Caltech101 for object recognition). Experimental results show that our approach outperforms many state-of-the-art algorithms even with a linear classifier.

Introduction

Bag-of-words (BOW) model is one of the most powerful and popular mid-level (Boureau et al. 2010) image representation models for image classification. This model represents an image as a histogram of visual words. The standard BOW-based framework is mainly composed of four steps: feature extraction, coding, spatial pooling and classification. The pipeline is almost fixed except for the "coding" part. Given an image, handcrafted low-level features, such as SIFT (Lowe 2004), HOG (Dalal and Triggs 2005) etc., are extracted from either interesting points or densely sampled patches. Such raw features are then encoded by various coding algorithms. In order to make the learned codes be robust to small transformations of the image, max-pooling is performed due to the fact that it has better performance than sum-pooling and average-pooling when paired with a linear classifier (Yang et al. 2009; Wang et al. 2010).

A good coding algorithm is the one that can encode similar patches with the similar codes, according to which, many elegant algorithms have been designed for the most important "coding" stage to improve the discriminative power of the learned codes in recent years. They vary from the traditional vector quantization (Lazebnik et al. 2006) to sparse coding (Wright et al. 2009; Yang et al. 2009), from hard-assignment to soft-assignment (Liu et al. 2011), from one-dictionary learning (Zhang and Li 2010; Mairal et al. 2009) to c-dictionary learning (Mairal et al. 2008) where c is the class number, and also from sparse coding to recently introduced low-rank coding (Chen et al. 2012; Zhang et al. 2013b; 2013a).

Sparse coding (SC) using ℓ_1 norm can encourage sparsity of the learned codes and is widely adopted currently. However, it has several limitations: firstly, as stated in (Wang et al. 2010), in order to favor sparsity, codes learned by using ℓ_1 norm may select quite different atoms for similar patches. Secondly, despite that many fast SC algorithms (Lee et al. 2006) have been proposed, the computation cost of ℓ_l norm is still very expensive. This hinders large-scale applications of SC algorithms. Thirdly, SC encodes each local points *independently*, which ignores the spatial consistency and contextual information of neighbor points.

Alternatively, low-rank coding (LRC) regularized by the trace-norm, can well handle the above problems. Generally speaking, since patches in the same image resemble mutually, the learned codes matrix should behave low-rankness property. In particular, as shown in (Liu et al. 2011; McCann and Lowe 2012), "local" methods[1] perform better than their "non-local" counterparts. Similarly, LRC encodes the local features jointly for coding consistency. In this sense, it is more "local" than SC. In addition, many efficient optimization algorithms have been proposed for LRC, such as singular value thresholding (SVT) (Cai et al. 2010). The computational bottleneck of SVT rests in the SVD decomposition of the data matrix. However, as is shown in (Zhang et al. 2013b; 2013a) and will be shown in the experiments, LRC is a good trade-off solution between the performance and encoding ef-

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]This refers to the algorithms which have considered the underlying manifold structure of local features.

ficiency.

Current LRC-based algorithms are incorporated with ℓ_1 norm (Zhang et al. 2013a; 2013b), which, as aforementioned, may deteriorate the performance of LRC because of favoring sparsity. Thus in this paper, we propose locality-constrained low rank coding (LCLR) which combines LRC with locality constraints (Yu et al. 2009; Wang et al. 2010). LCLR enjoys joint coding and locality, such that LCLR exploits the manifold structure of local features in a more thorough manner apart from simply adding a low-rank regularization term. Codes are efficiently learned by the proposed online algorithm based on the inexact augmented Lagrange multiplier method. In the experiments, we show that by taking advantages of spatial layout of local features, the learned representations are more discriminative than that of many state-of-the-art algorithms for various classification tasks.

Low-rank Representations

Consider a data matrix $X = [x_1, x_2, ..., x_n]$ in $\mathbb{R}^{d\times n}$, where n is the number of samples. Each column x_i is a feature vector of dimension d and can be represented by a linear combination of the atoms from an overcomplete dictionary $B = [b_1, b_2, ..., b_k]$:

$$X = BZ, \tag{1}$$

where $Z = [z_1, z_2, ..., z_n]$ is the codes (or coefficients) of X. Since the dictionary is often overcomplete ($k > d$), the above equation is an ill-posed problem and can have multiple feasible solutions. To solve Eqn. 1, various regularization terms have been presented, including the famouse ℓ_1 norm (Candes et al. 2006; Wright et al. 2009) and trace-norm (Liu et al. 2010)[2].

Mathematically, low-rank coding is that we search for a low-rank representation Z of X by solving

$$\min_Z rank\left(Z\right), \text{s.t. } X = BZ. \tag{2}$$

Using the Lagrange multiplier method, the above equation can be transformed to its unconstrained version under the least square approximation:

$$\min_Z rank\left(Z\right) + \lambda \left\|X - BZ\right\|_F^2, \tag{3}$$

where λ is the lagrange multiplier and $\left\|\cdot\right\|_F$ is the Frobenius norm. However, due to the discreteness of the $rank$ function, direct optimization of Eqn. 3 is NP-hard. Fortunately, Candes et al. (Candes and Plan 2010) show that the trace-norm is a good surrogate for the $rank$ function, leading to the following equation:

$$\min_Z \left\|Z\right\|_* + \lambda \left\|X - BZ\right\|_F^2, \tag{4}$$

where $\left\|\cdot\right\|_*$ is the trace-norm (or nuclear-norm, i.e., the sum of the singular values). Zhang et al. (Zhang et al. 2013a) add a sparsity inducing term $\left\|Z\right\|_{1,1} = \sum_{i=1} \left\|z_i\right\|_1$ into Eqn. 4 because sparse feature coding has been shown to be quite helpful in image classification (Wright et al. 2009; Yang et al. 2009).

[2]We will omit here the general formulation of sparse coding due to the limited space.

Alternatively in the literature of face recognition, in order to get a more robust representation of X, a common hypothesis is that $X = BZ + E$, where E is the noise matrix. Thus a more general rank minimization problem can be formulated as:

$$\min \left\|Z\right\|_* + \lambda \left\|E\right\|_1, \text{s.t. } X = BZ + E. \tag{5}$$

Based on Eqn. 5, Zhang et al. (Zhang et al. 2013b) trains for each class a sub-dictionary to increase the discrimination power of each class-specific sub-dictionary B_i. Specifically, they update the concatenated dictionary (e.g., $B = [B_1, B_2, ..., B_c]$ where c is the class number) as a whole such that a discriminative dictionary B can be learned from all training sample simultaneously. This is much faster than updating the sub-dictionary one by one (Chen et al. 2012). Similar with (Zhang et al. 2013a), they also add a sparsity inducing term $\left\|Z\right\|_1$ to do atom selection. Moreover, they associate label information (Huang et al. 2013) in dictionary learning to increase the discriminative power of the learned codes.

Locality-constrained Low-rank Coding (LCLR)

Following the general BOW pipeline (Liu et al. 2011; Yang et al. 2009; Zheng et al. 2011), LCLR splits an image into m same-sized and overlapped patches and extracts dense SIFT features (Lazebnik et al. 2006) from them. These m feature vectors are concatenated in a matrix $X = [x_1, x_2, ..., x_m] \in \mathbb{R}^{128\times m}$. Since the low-rankness is more prominent within the same image, LCLR performs low-rank coding on each image (e.g., X). Basically speaking, LCLR is based on the following two observations:

- Given a dictionary $B = [B_1, B_2, ..., B_k] \in \mathbb{R}^{128\times k}$, where k is the number of atoms, similar descriptors should have similar representations w.r.t. B. In other words, similar descriptors should choose similar atoms, i.e., the learned representation Z should be low-rank.
- Inspired by (Yu et al. 2009; Wang et al. 2010) and the discussions in the above section, locality is more essential than sparsity. Therefore, sparse feature coding using ℓ_1 norm may not be the optimal choice.

In order to further exploit spatial layout of X and increase the coding consistency of current low-rank based algorithms, we incorporate a locality-constrained regularization term (Wang et al. 2010) into Eqn. 4, and the final criteria can be formulated as:

$$\min_{Z_1,Z_2} \frac{1}{2} \left\|X - BZ_1\right\|_F^2 + \lambda_1 \left\|Z_1\right\|_* + \frac{\lambda_2}{2} \sum_{i=1}^{m} \left\|d_i \odot z_{1,i}\right\|_2^2, \tag{6}$$

where z_i is the i-th column of Z, $\odot$ stands for element-wise multiplication, and

$$d_i = \exp\left(\frac{\text{dist}\left(x_i, B\right)}{\sigma}\right), \tag{7}$$

where $\text{dist}\left(x_i, B\right) = \left[\text{dist}\left(x_i, b_1\right), \ldots, \text{dist}\left(x_i, b_k\right)\right]^{\mathrm{T}}$, and $\text{dist}\left(x_i, b_j\right)$ is the Euclidean distance between x_i and b_j. σ

controls the bandwith of the distribution. Furthermore, d_i is normalized to be within the range of $(0, 1]$ by substracting $\max(\text{dist}(x_i, B))$ from $\text{dist}(x_i, B)$.

Optimization

To solve Eqn. 6, we first convert it to the following equivalent format:

$$\min_{Z_1,Z_2} \tfrac{1}{2}\|X - BZ_1\|_F^2 + \lambda_1\|Z_2\|_* + \tfrac{\lambda_2}{2}\sum_{i=1}^{m}\|d_i \odot z_{1,i}\|_2^2, \quad \text{s.t. } Z_1 = Z_2, \tag{8}$$

where an auxiliary variable Z_2 is introduced to make the objective function seperable. Using the Augmented Lagrange Multiplier method, we can derive

$$\min_{Z_1,Z_2} \tfrac{1}{2}\|X - BZ_1\|_F^2 + \lambda_1\|Z_2\|_* + \tfrac{\lambda_2}{2}\sum_{i=1}^{m}\|d_i \odot z_{1,i}\|_2^2 + \text{trace}\left(Y_1^{\mathrm{T}}(Z_1 - Z_2)\right) + \tfrac{\mu}{2}\|Z_1 - Z_2\|_F^2, \tag{9}$$

where Y_1 is lagrange multiplier and $\mu > 0$ is a penalty parameter. The optimization of Eqn. 9 can be divided into two subproblems. The first is to update the codes Z while fixing the dictionary B. The other is to update the dictionary B for the given codes Z.

Updating Z Given B Given B, we can update the remaining variables (e.g., Z_1 and Z_2) one by one. The scheme is as follows:

$$\begin{aligned} Z_2^* &= \arg\min \lambda_1\|Z_2\|_* + \text{trace}\left(Y_1^{\mathrm{T}}(Z_1 - Z_2)\right) + \tfrac{\mu}{2}\|Z_1 - Z_2\|_F^2 \\ &= \arg\min \tfrac{\lambda_1}{\mu}\|Z_2\|_* + \tfrac{1}{2}\|Z_2 - G\|_F^2, \end{aligned} \tag{10}$$

where $G = Z_1 + \frac{1}{\mu}Y_1$.

$$\begin{aligned} z_{1,i}^* &= \arg\min \tfrac{1}{2}\|X - BZ_1\|_F^2 + \tfrac{\lambda_2}{2}\sum_{i=1}^{m}\|d_i \odot z_{1,i}\|_2^2 \\ &\quad + \text{trace}\left(Y_1^{\mathrm{T}}(Z_1 - Z_2)\right) + \tfrac{\mu}{2}\|Z_1 - Z_2\|_F^2 \\ &= \arg\min \tfrac{1}{2}\sum_{i=1}^{m}\|x_i - Bz_{1,i}\|_2^2 + \lambda_2\left\|d_i^{\mathrm{T}} z_{1,i}\right\|_2^2 \\ &\quad + 2Y_{1,i}^{\mathrm{T}}(z_{1,i} - z_{2,i}) + \mu\|z_{1,i} - z_{2,i}\|_2^2 \\ &= \left(B^{\mathrm{T}}B + \lambda_2 d_i^{\mathrm{T}} d_i + \mu\mathrm{I}\right)^{-1}\left(B^{\mathrm{T}}x_i + \mu z_{2,i} - Y_{1,i}\right), \end{aligned} \tag{11}$$

in which $z_{1,i}$ is the i-th item of Z_1, i.e., $Z_1 = [z_{1,1}, ..., z_{1,i}, ..., z_{1,m}]$. However, since the optimization of Z_1 is performed item-wise (see the above equation), the computation cost will be huge especially when m is large. Therefore, we propose an online strategy to solve this problem. Specifically, we randomly choose n data points from X, where $n \ll m$, and update Z_1 by

$$Z_1^* = \left(B^{\mathrm{T}}B + \lambda_2 d_i^{\mathrm{T}} d_i + \mu\mathrm{I}\right)^{-1}\left(B^{\mathrm{T}}X + \mu Z - Y_1\right), \tag{12}$$

where $1 \le i \le n$. In this way, Z_1 can be updated as a whole. In the experiments, we found that this online step converges to the similar point as the "item-wise" step (Eqn. 11), while spending much less time. Most importantly, it performs surprisingly well.

Algorithm 1 online coding by LCLR

Input: data matrix $X \in \mathbb{R}^{128\times m}$, dictionary B, and parameters μ, λ_1, λ_2, σ
Output: learned codes Z_1, Z_2

```
Initialize Z_1 = Z_2 = Y_1 = 0, ε = 10^-6, μ_max = 10^30, ρ = 1.3, maxiter = 100, index = randperm(m)
for j = 1 : m do
    id = index[j]; x_i ← X[id];
    Fix Z_1 and update variable Z_2 according to Eqn. 10
    Fix Z_2 and update variable Z_1 according to Eqn. 11
    Update the multiplier:
        Y_1^{j+1} = Y_1^j + μ^j (Z_1^j − Z_2^j)
    Update the parameter μ:
        μ^{j+1} = min(ρμ^j, μ_max)
    Check the convergence condition
        ||Z_1^{j+1} − Z_2^{j+1}||_∞ ≤ ε or j > maxiter
    if converged then
        return Z_1, Z_2
    end if
end for
```

Note that the optimization problem of Eqn. 10 can be efficiently solved by a singular value shrinkage operator (Cai et al. 2010):

$$S_{\frac{\lambda_1}{\mu}}(G) = U\,\text{diag}\left(\left\{\sigma_i - \frac{\lambda_1}{\mu}\right\}_+\right)V^{\mathrm{T}}, \tag{13}$$

where $G = U\Sigma V^{\mathrm{T}}$, $\Sigma = \text{diag}(\sigma_i)$, $i = 1...r$ is the SVD decomposition of a matrix G of rank r , and $\{t\}_+$ is the positive part of t. We conclude the LCLR coding algorithm in Algorithm 1. Note that n, equaling to the final value of j, is determined automatically.

Updating B Given Z To update the dictionary B, we can rewrite Eqn. 9 as the following formula by retaining all the variables associated with B,

$$B^* = \arg\min \frac{1}{2}\|X - BZ_1\|_F^2 + \frac{\lambda_2}{2}\sum_{i=1}^{m}\|d_i \odot z_{1,i}\|_2^2. \tag{14}$$

As suggested by (Wang et al. 2010), when optimizing the above equation, we simply drop the last term leading to

$$B^* = \arg\min \frac{1}{2}\|X - BZ_1\|_F^2. \tag{15}$$

This is of the form of minimizing a differentiable function over a closed convex set. We use gradient descent for solving the above problem. The updating scheme is as

$$\begin{aligned} B^{j+1} &= B^j - \beta\nabla B^j, \\ \nabla B^j &= -2\left(X - B^j Z_1\right)Z_1^{\mathrm{T}}, \end{aligned} \tag{16}$$

where β is the step length controlling the learning rate. For completeness, we present here the dictionary learning algorithm in Algorithm 2, where the pre-given parameter k is the number of atoms in the dictionary.

Algorithm 2 dictionary learning by LCLR

Input: data matrix X, and parameters μ, λ_1, λ_2, k
Output: learned codes Z and the dictionary B

Initialize $\epsilon = 10^{-6}$, $maxiter = 100$, and B^0 by K-means algorithm
while not converged **do**
  Optimize for Z_1, Z_2 while fixing B^0 by Algrithm 1
  Fix Z_1, Z_2 and update B according to Eqn. 16
  Check the convergence condition
$$\left\|B^{j+1} - B^j\right\|_\infty \leq \epsilon$$
  if converged **then**
    return B^{j+1}
  end if
end while

LCLR Revisiting

As stated earlier, low-rank coding is recently introduced for image classification. In (Zhang et al. 2013a), Zhang et al. propose low-rank sparse coding (LRSC) which incorporates low-rank coding with sparse coding. Specifically, LRSC uses superpixel generating algorithms (e.g., SLIC(Achanta et al. 2010)) on each image and imposes low-rankness on those segmented regions. This is for the purpose of efficiency since do SVD on small matrix is extremely fast. However, this fails to capture the spatial consistency of the whole image. In addition, as aforementioned, using ℓ_1 norm may select quite different atoms for similar patches, which may further deteriorate the performance of LRSC. On the other side[3], Chen et al. (Chen et al. 2012) train for each class a sub-dictionary which is updated one-at-a-time. They also add a regularization term to decrease the representation power of B_i to class j ($j \neq i$). More recently, Zhang et al. (Zhang et al. 2013b) train a well-structured dictionary such that all the sub-dictionaries can be updated simultaneously. However, we argue that these two methods are either computationally expensive or memory-consuming.

LCLR combines the advantages of LRC and locality-constrained coding, and at least have the following three properties: a) If we set λ_1 to zero, then LCLR falls back to the original LLC (Wang et al. 2010). When $\lambda_2 \neq 0$, LCLR generates sparse codes in the sense that the solution only has few significant values. From this aspect, LCLR plays the role of atom selection the same as ℓ_1 norm. b) LCLR is a more "local" method than other coding paradigms. Compared to SC based methods, LCLR adopts trace-norm which takes spatial consistency and contextual information into consideration; Compared to LRC based methods, it adopts locality-constrained regularization term rather than ℓ_1 norm. c) The optimization of LCLR is very efficient and the solutions of LCLR can all be derived analytically.

Experiments

We evaluate LCLR on 4 challenging benchmarks, including one face dataset (Extended YaleB (Georghiades et al. 2001)), one handwritten dataset (USPS[4]), and two image datasets (Caltech101 (Fei-Fei et al. 2007) for object recognition and Scene13 (Fei-Fei and Perona 2005) for scene recognition). We compare LCLR with several state-of-the-art algorithms on each dataset to show the effectiveness. In each experiment, we keep all the steps the same as that of the baselines except for the coding stage for fair comparison.

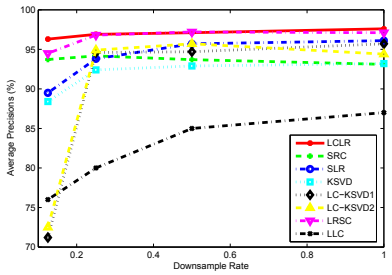

Figure 1: Performance comparison on Extended YaleB

Face Recognition

The Extended YaleB dataset, containing 2,414 frontal face images of 38 people, is widely adopted in face recognition (FR). For each person, there are about 64 images. This dataset is challenging because each image is taken under various lighting conditions. The cropped and normalized face images have size $192 \times 168 = 32,256$ pixels. Following the general setup, we test LCLR on the original images and down-sampled images, leading to feature dimensions of $96 \times 84 = 8064$, $48 \times 42 = 2016$, and $24 \times 21 = 504$. Instead of using FR-specific features (such as Randomface (Wright et al. 2009), Eigenface (Turk and Pentland 1991), etc.), we simply use the raw pixels as features for better illustrating the advantages of the proposed coding scheme. The parameter settings are: $\lambda_1 = 1, \lambda_2 = 500, \sigma = 200, \mu = 10^{-2}, k = 512$. We randomly select 32 images for training with the remaining images for testing. Each experiment is repeated 10 times and the average precision is reported.

We compare LCLR with two recently proposed low-rank based algorithms: LRSC (Zhang et al. 2013a) and SLR (Zhang et al. 2013b), and 4 classic algorithms which reported state-of-the-art results on this dataset, i.e., SRC (Wright et al. 2009), K-SVD (Aharon et al. 2005), FDDL (Yang et al. 2011), and LC-KSVD (Jiang et al. 2011). Since the source codes of LRSC and SRC are not publicly available, results reported here are based our implementations. Parameters are set according to their original papers. When implementing SRC, we use the "solveLasso" function in the package provided by SparseLab [5]. The comparative results are shown in Figure 1, where LC-KSVD2 differs from LC-KSVD1 in that it considers classification error. It can be observed that by considering structure information, low-rank based algorithms (i.e., LCLR, LRSC, and SLR) outperform other baselines. Moreover, we find that LCLR and LRSC

[3]This stands for methods evolving from Eqn. 5

[4]http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

[5]http://sparselab.stanford.edu/

Table 1: Runtime of different coding methods on Extended YaleB where "Tr, Te, and Ttl" stand for "Training ,Testing, and Total" respectively.

Methods	Tr Time (ms)	Te Time (ms)	Ttl Time (m)
LCLR	16.8	19.5	0.73
LRSC	12.3	12.6	0.50
FDDL	2697.4	2350.8	101.6
LC-KSVD1	89.86	0.38	1.83
LC-KSVD2	90.97	0.36	1.85
SRC	0	382.5	7.64
LLC	0.57	0.6	0.02

Table 2: Recognition accuracy on Extended YaleB.

Methods	SRC	DLSI*	FDDL	LRSC	LCLR
Acc. (%)	90	89	91.9	92.1	**94.6**

perform consistently better than SLR. This may due to the fact that restoring X by BZ using Frobenius norm (Eqn. 3) is more discriminative than ℓ_1 norm (Eqn. 5).

We also evaluate the computation cost of various coding algorithms. In this experiment, all images are resized to 24×21, and 32 randomly selected images are used for training for each person. The training time is the average time over the entire training set, and the testing time, including both coding and classification time, is averaged over the entire testing set. As is shown in Table 1, LLC is the fastest algorithm, while LRSC and LCLR are close behind. SRC uses training samples directly as the dictionary which omits the dictionary learning process. However, the testing phase is time-consuming. LC-KSVD can efficiently classify a given image while the training cost is not trivial. In general, LCLR achieves the best performance while keeping the time cost satisfactory. All experiments are carried out using MATLAB on a Intel Core i7-4770K PC with 16GB RAM.

As shown in Table 1, FDDL is computationally expensive, thus we did not include FDDL in Figure 1. In order to compare LCLR with FDDL, we implement LCLR as well as LRSC under FDDL's experimental settings (Yang et al. 2011), and the comparative results are shown in Table 2. All results reported in Table 2 are copied from (Yang et al. 2011) except for LCLR and LRSC. Again, LCLR and LRSC outperform all the methods and LCLR makes an improvement of 2.5% accuracy over LRSC.

Handwritten Digits Recognition

The widely used USPS dataset is adopted for this task. USPS contains 9,298 images (7,291 for training and 2,007 for testing) of 10 handwritten digits, i.e., $0 \sim 9$. We compare LCLR with LRSC (Zhang et al. 2013a) as well as algorithms which reported state-of-the-art results on this dataset, including GraphSC (Zheng et al. 2011), FDDL (Yang et al. 2011), and SDL (Mairal et al. 2009) and some problem-specific methods (such as SVM-Gauss, k-NN). GraphSC assumes that if two data points x_i and x_j are similar, then z_i and z_j should be similar too. SDL increases the discriminative power of the learned codes by merging coding and classification into one single step. Here the original images of size 16×16 are directly used as the feature, and the parameter settings are the same as those for Extended YaleB except that $k = 1024$ and $\rho = 1.1$.

Table 3: Error rates on the USPS dataset.

Methods	Error Rate (%)
LCLR	**1.14**
LRSC	6.20
SDL-D	3.54
FDDL	3.69
GraphSC	5.00
k-NN	5.20
SVM-Gauss	4.20

Table 3 lists the results of LCLR and other baselines. Except for LCLR and LRSC, all remaining results are copied from (Yang et al. 2011). As is shown in Table 3, LCLR perform much better than (boosts nearly 2.5% over the second best method) all other baselines. It is worth noting that the best error rate published on this dataset is 2.4% (Haasdonk and Keysers 2002), using the method tailored to this task. Our result also defeats theirs.

Object and Scene Recognition

Other important applications of various coding algorithms are object and scene recognition, which attracts much attention over decades. We follow the standard experimental settings as that of methods used as baselines (Wang et al. 2010; Yang et al. 2009; Liu et al. 2011): 1) All images are downsized to no more than 300×300 pixels. 2) Dense SIFT features (Lazebnik et al. 2006) are extracted from all images from a single scale of 16×16 patches with an 8 pixel stepsize. 3) In the experiments, we found that the dictionary update algorithm (e.g. Algorithm 2) only improves the performance slightly and is time-consuming. Thus we simply use K-means algorithm to generate the dictionary, and use this for Algorithm 1. 4) Max-pooling is adopted as it performs better than sum-pooling (Wang et al. 2010; Liu et al. 2011). In order to incorporate spatial information, SPM kernel (Lazebnik et al. 2006) with 3 levels of 1×1, 2×2, and 4×4 is utilized. 5) Linear-SVM (Fan et al. 2008) is chosen for the classification purpose.

Scene-13 Scene-13 (Fei-Fei and Perona 2005) contains 3,859 images of 13 classes. For each class, there are $200 \sim 400$ images. These images include indoor environments and outdoor scenes. Following the standard protocol, we randomly choose 100 images from each class for training and the rest for testing. We compare LCLR with SPM (Lazebnik et al. 2006), LLC (Wang et al. 2010), ScSPM (Yang et al. 2009), LSC (Liu et al. 2011), and SC (Huang et al. 2011). Experiments are repeated 5 times and the averaged results are shown in Table 4. Table 4 is divided into two sections. The bottom one lists the results of LCLR, as well as the

Table 4: Recognition accuracy on the Scene-13 dataset.

Methods	Accuracy (%)	Methods	Accuracy (%)
HC	77.20	ScSPM	83.14
LLC	83.25	LSC	83.33
SC	82.11	LRSC	**85.13**
LCLR	**83.86**	LRSC	82.81

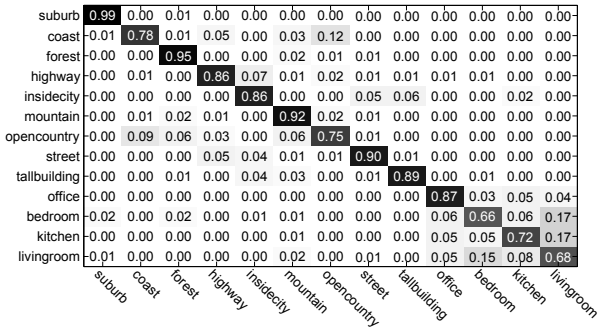

Figure 2: The confusion matrix of the Scene-13 dataset.

LRSC implemented by ourselves, and the top part lists the results of all baselines copied from (Zhang et al. 2013a). The performance of LRSC decreases because in (Zhang et al. 2013a), in order to better demonstrate the power of low-rank coding, they use a 4 (which is 8 in our setting) pixel stepsize. However, the denser the patches are sampled, the better results the algorithm can obtain. As shown, our method attains higher accuracy than LRSC under the same settings, and outperform other baselines except for LRSC even if they have denser feature points. We further show the confusion matrix in Figure 2, from which we can see that "bedroom" and "livingroom" are misclassified most severely. We expect LCLR to perform better if more features can be extracted, because for these two classes, the SIFT descriptor looks similar.

Caltech101 The Caltech101 dataset (Fei-Fei et al. 2007) contains 9,144 images from 102 classes (i.e., 101 object classes as well as a "background" class). Samples from the same class have significant shape variability. The number of images in each class varies from 31 to 800. Following the common experimental settings, we train on 5, 10, 15, 20, 25 and 30 images per class and test on the rest.

We compare LCLR with two low-rank based algorithms, LRSC (Zhang et al. 2013a) and SLR (Zhang et al. 2013b) as well as several state-of-the-art algorithms, including KSVD (Aharon et al. 2005), D-KSVD (Zhang and Li 2010), LLC (Wang et al. 2010), SRC (Wright et al. 2009), LC-KSVD (Jiang et al. 2011) and ScSPM (Yang et al. 2009). The comparative results are shown in Table 5. We divide the table into two sections, where the bottom part lists three low-rank coding (LRC) based algorithms. It can be observed that by

Table 5: Recognition accuracy on the Caltech101 dataset.

# of Tr. Images	5	10	15	20	25	30
ScSPM	-	-	67.0	-	-	73.2
SRC	48.8	60.1	64.9	67.7	69.2	70.7
LLC	51.2	59.8	65.4	67.7	70.2	73.4
KSVD	49.8	59.8	65.2	68.7	71.0	73.2
D-KSVD	49.6	59.5	65.1	68.6	71.1	73.0
LC-KSVD2	54.0	63.1	**67.7**	70.5	72.3	73.6
SLR	-	-	66.1	-	-	73.6
LRSC	**55.0**	**63.5**	67.1	70.3	72.7	74.4
LCLR	53.4	62.8	67.2	**70.8**	**72.9**	**74.7**

Figure 3: Example images from "metronome", "pagoda" and "minaret" (from top to bottom) with 100% accuracy.

taking advantages of structural information and joint coding, LRC-based methods win 5 times in all 6 sets of statistics. Moreover, LCLR achieves increasingly better performance when more and more training images are used. Finally, when using 30 training images, LCLR get the accuracy of 74.7% which is also comparative to the result (75.02%) reported in the original paper of LRSC (Zhang et al. 2013a). However, we expect LCLR to perform better than theirs if LCLR uses 4 pixels stepsize when sampling patches.

There are a total of 9 classes achieving 100% recognition accuracy when using 30 training images per class. These classes are "accordion", "leopards", "car", "garfield", "metronome", "minaret", "pagoda", "scissors" and "snoopy". Figure 3 shows some sample images from three of these classes which have the most similar shape. This demonstrates the discriminative power of LCLR for similar objects.

Conclusions

In this paper, we present a new coding strategy, locality-constrained low-rank algorithm (LCLR), for image classification. To solve the computational bottleneck of LCLR, we also propose an efficient online optimization algorithm. LCLR, which imposes coding consistency by joint coding and locality-constraints, exploits the underlying manifold of data space and dictionary space in a more thorough manner than current low-rank based algorithms. Extensive experiments show that our method improves the state-of-the-art results on several benchmarks with only a linear classifier.

Acknowledgement

The research work described in this paper was fully supported by the grants from the National Natural Science Foundation of China (Project No. 61375045) and Beijing Natural Science Foundation (Project No. 4142030).

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2010. Slic superpixels. *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep* 149300.

Aharon, M.; Elad, M.; and Bruckstein, A. 2005. K-svd: Design of dictionaries for sparse representation. *Proceedings of SPARS* 5:9–12.

Boureau, Y.-L.; Bach, F.; LeCun, Y.; and Ponce, J. 2010. Learning mid-level features for recognition. In *CVPR*, 2559–2566.

Cai, J.-F.; Candès, E. J.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4):1956–1982.

Candes, E. J., and Plan, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE* 98(6):925–936.

Candes, E. J.; Romberg, J. K.; and Tao, T. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics* 59(8):1207–1223.

Chen, C.-F.; Wei, C.-P.; and Wang, Y.-C. 2012. Low-rank matrix recovery with structural incoherence for robust face recognition. In *CVPR*, 2618–2625.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, 886–893.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR* 9:1871–1874.

Fei-Fei, L., and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, 524–531.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding* 106(1):59–70.

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI* 23(6):643–660.

Haasdonk, B., and Keysers, D. 2002. Tangent distance kernels for support vector machines. In *ICPR*, volume 2, 864–868.

Huang, Y.; Huang, K.; Yu, Y.; and Tan, T. 2011. Salient coding for image classification. In *CVPR*, 1753–1760.

Huang, J.; Nie, F.; Huang, H.; and Ding, C. 2013. Supervised and projected sparse coding for image classification. In *AAAI*, 438–444.

Jiang, Z.; Lin, Z.; and Davis, L. S. 2011. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR*, 1697–1704.

Lazebnik, S.; Schmid, C.; and Ponce, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, 2169–2178.

Lee, H.; Battle, A.; Raina, R.; and Ng, A. 2006. Efficient sparse coding algorithms. In *NIPS*, 801–808.

Liu, G.; Lin, Z.; and Yu, Y. 2010. Robust subspace segmentation by low-rank representation. In *ICML*, 663–670.

Liu, L.; Wang, L.; and Liu, X. 2011. In defense of soft-assignment coding. In *ICCV*, 2486–2493.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60(2):91–110.

Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; and Zisserman, A. 2008. Discriminative learned dictionaries for local image analysis. In *CVPR*, 1–8.

Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; and Zisserman, A. 2009. Supervised dictionary learning. In *NIPS*.

McCann, S., and Lowe, D. G. 2012. Local naive bayes nearest neighbor for image classification. In *CVPR*, 3650–3656.

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of cognitive neuroscience* 3(1):71–86.

Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; and Gong, Y. 2010. Locality-constrained linear coding for image classification. In *CVPR*, 3360–3367.

Wright, J.; Yang, A. Y.; Ganesh, A.; Sastry, S. S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *TPAMI* 31(2):210–227.

Yang, J.; Yu, K.; Gong, Y.; and Huang, T. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 1794–1801.

Yang, M.; Zhang, L.; Feng, X.; and Zhang, D. 2011. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 543–550.

Yu, K.; Zhang, T.; and Gong, Y. 2009. Nonlinear learning using local coordinate coding. In *NIPS*, 2223–2231.

Zhang, Q., and Li, B. 2010. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2691–2698.

Zhang, T.; Ghanem, B.; Liu, S.; Xu, C.; and Ahuja, N. 2013a. Low-rank sparse coding for image classification. In *ICCV*, 281–288.

Zhang, Y.; Jiang, Z.; and Davis, L. S. 2013b. Learning structured low-rank representations for image classification. In *CVPR*, 676–683.

Zheng, M.; Bu, J.; Chen, C.; Wang, C.; Zhang, L.; Qiu, G.; and Cai, D. 2011. Graph regularized sparse coding for image representation. *TIP* 20(5):1327–1336.