From Dictionary of Visual Words to Subspaces: Locality-constrained Affine Subspace Coding *

Peihua Li, Xiaoxiao Lu, Qilong Wang
School of Information and Communication Engineering, Dalian University of Technology
peihuali@dlut.edu.cn, shaw,qlwang@mail.dlut.edu.cn

Abstract

The locality-constrained linear coding (LLC) is a very successful feature coding method in image classification. It makes known the importance of locality constraint which brings high efficiency and local smoothness of the codes. However, in the LLC method the geometry of feature space is described by an ensemble of representative points (visual words) while discarding the geometric structure immediately surrounding them. Such a dictionary only provides a crude, piecewise constant approximation of the data manifold. To approach this problem, we propose a novel feature coding method called locality-constrained affine subspace coding (LASC). The data manifold in LASC is characterized by an ensemble of subspaces attached to the representative points (or affine subspaces), which can provide a piecewise linear approximation of the manifold. Given an input descriptor, we find its top-k neighboring subspaces, in which the descriptor is linearly decomposed and weighted to form the first-order LASC vector. Inspired by the success of usage of higher-order information in image classification, we propose the second-order LASC vector based on the Fisher information metric for further performance improvement. We make experiments on challenging benchmarks and experiments have shown the LASC method is very competitive.

1. Introduction

Encoding methods as one key component of Bag of Visual Words (BoVW) model [34] have made great progress in the past years. Such coding methods achieve impressive results in many computer vision tasks, such as object recognition, image retrieval, and texture classification. From the hard-assignment approaches [34, 6, 22] to the soft-assignment ones [42, 36, 24], and then to those based on local constraints [45, 38] and the methods using higher order information [18, 47, 44, 31], the advance of encoding methods has been playing a great role in improving the classification performance [16]. This is mainly due to consideration of the algebraic and geometric structure of data manifold and utilization of high-order information.

The sparsity of coding vector is an inherent property in the BoVW model. The hard-assignment methods [34, 6, 22] allocate each feature to the nearest word in the dictionary, producing extremely sparse vector which has only one nonzero component. The sparse coding (SC) method [42] representing each feature as a sparse, linear combination of visual words, having smaller reconstruction error and achieving better performance than the hard-assignment methods. However, the SC method is computationally demanding; moreover, the non-smooth ℓ_1 regularizer introduces the negative effect [38] that quite different words may be selected for similar patches to favor sparsity, leading to loss of correlation between the corresponding coding vectors.

Recent research has observed and validated that locality is more essential than sparsity [38, 45, 24]. The locality-constrained linear coding (LLC) [38] is a great advance in this aspect, achieving state-of-the-art recognition performance. The LLC method imposes the weighted ℓ_2 regularizer in the least square-based cost function where the weights are proximity measures of the feature to individual words. The LLC method yields the local smooth sparsity which ensures similar patches have similar encoding vectors. In addition, it has closed-form solution such that it is faster than the SC methods. A most often used form of LLC is to explicitly represent the feature as a linear combination of the most nearby visual words, and by doing so the LLC method becomes much more efficient.

In the BoVW methods we usually face high-dimensional feature space (e.g. the dimension of SIFT features is 128 [25]). It is known that the feature data are often located on some low-dimensional manifold and leveraging the geometry of the manifold may bring benefits [30]. As shown in Figure 1 (a), the LLC method characterizes the geometry of feature space by some representative points (visual words), obtained by either k-means or dictionary learning methods [42, 38]. The dictionary thus obtained only provides crude, piecewise constant approximation of the manifold [3]. Be-

*The work was supported by the NSFC (61471082).

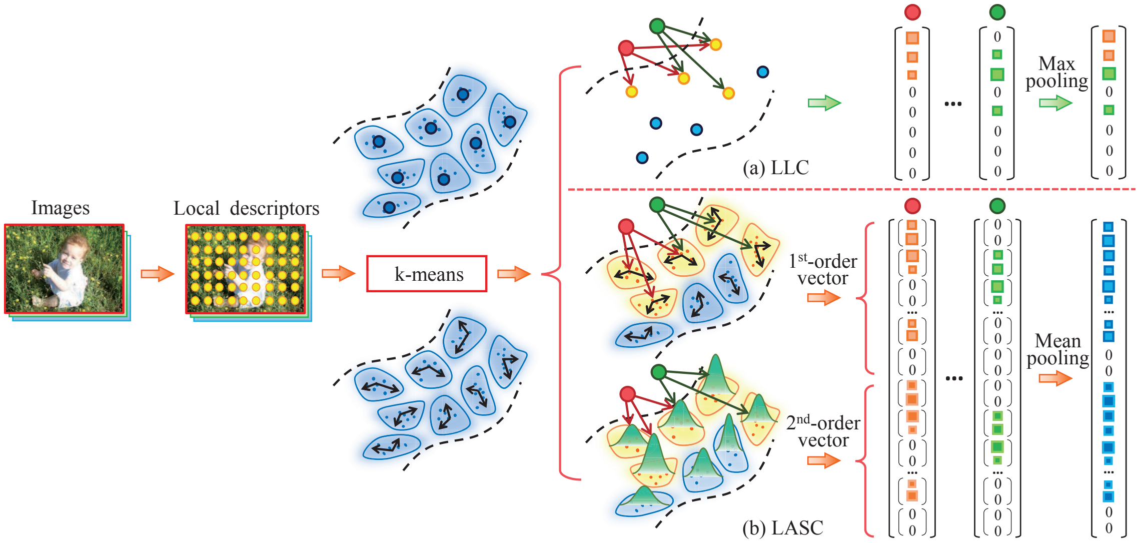

Figure 1. Locality-constrained affine subspace coding (LASC). The dictionary of LLC (a) is a set of representative points (visual words); the geometric structure immediately surrounding the words are discarded, and so it only provides a crude, piecewise constant approximation of the manifold [3]. In contrast, the dictionary of LASC (b) is an ensemble of low-dimensional linear subspaces attached to the representative points (affine subspaces), which provides a piecewise linear approximation of the data manifold [3]. For an input feature, we find its top-k nearest subspaces and perform linear decomposition of the feature in these subspaces weighted by the proximity measures. Beyond the linear coding, we propose to leverage the second-order information of the descriptors based on the Fisher information metric.

cause the LLC method discards the geometric structure immediately surrounding each representative point, it has limited capability to characterize the feature distribution on the data manifold. Can this problem be addressed by increasing the number M of visual words? Recent work [16] has shown that by increasing M the classification performance can improve but may saturate to the upper bound [1]. This demonstrates that the increase of M alleviates but cannot address this problem. This phenomenon is not surprising because (1) all feasible training samples are sparsely populated in high dimensions due to the curse of dimensionality [15, Chap. 2], and it is difficult, if not impossible, to obtain sufficiently large size of training samples because of restriction of memory and computing capability, and (2) even one can approximate the feature space with much more visual words, the geometric structure surrounding these words are still not considered.

We approach this problem by presenting a novel encoding method called Locality-constrained Affine Subspace Coding (LASC). Figure 1 shows the flowchart of the LASC method and comparison with LLC. We explicitly model the geometric structure of the immediate neighborhoods of the representative points by low-dimensional linear subspaces [30, 19], which indeed provides a piecewise linear approximation of the manifold underlying data [3]. Hence, the dictionary of LASC is an ensemble of low-dimensional linear subspaces attached to the representative points, or affine subspaces. Our idea is to perform descriptor encoding only in few most neighboring subspaces. The neighbor indicates some metric and we introduce the proximity measures between points and subspaces from the perspectives of statistical learning and reconstruction error. Specifically, for a given descriptor, we find its top-k nearest affine subspaces and perform linear decomposition in these subspaces weighted by the proximity measure. This way, we produce the first-order (linear) LASC vector of the descriptor. Motivated by the success of higher-order encoding method, we propose the second-order LASC vector based on the Fisher information metric to further improve the performance of the proposed LASC method.

The rest of this paper is organized as follows. We introduce in § 2 the related work. We then describe in detail the proposed LASC method in § 3. The experiments are presented in § 4. Finally, § 5 gives the concluding remark.

2. Related Work

The local coordinate coding (LCC) [45] learned a nonlinear function in high dimensional feature space by considering the geometric structure of the data. Recent work [43] described a mixture sparse coding model which can be regarded as an approximate model of LCC. The local tangent coding (LTC) [44] extended LCC by introducing local tangent directions computed by the principal component analysis (PCA). Super vector (SV) coding [47] is a spe-

[1] It has been observed [16] that the classification performance of LLC saturated on PASCAL VOC 2007 around $M = 32,768$.

cial case of LTC, which, inheriting the main characteristics of LTC but running much faster, has been widely used in image classification [5, 16]. The proposed method is similar to LTC, but has several significant differences. First, the LASC method is intended to perform encoding on an ensemble of affine subspaces as opposed to the encoding of LTC on individual visual words. Compared to the accurate approximation to the nonlinear function in the LTC method, the LASC method focuses on highly distinct representation. Moreover, we find k most neighboring affine subspaces using the proximity measures between features and affine subspaces to weight the coding vector. It is basically different from LTC which computes the LCC coefficients of the feature on the visual words to weight the code by solving the LASSO problem [15]. In contrast with LTC, the LASC throughout has closed-form and is much more efficient. Last but not the least, we present the second-order encoding in each subspace based on the Fisher Information metric [31], which amounts to explore the geometry of the Riemannian manifold from the statistic perspective [17]. It has been shown that leveraging higher-order statistics of features benefits greatly the classification tasks, and such kind of works include the second-order pooling method [4], VLAD-based method using higher-order statistics [28] and those based on the fisher vector (FV) [31, 21, 27].

Nandakishore *et al.* [19] propose to find the nearest subspace in a set of ones for dimensionality reduction. The difference is that instead of dimensionality reduction in [19], we are interested in higher dimensional representation which has great distinctiveness. We notice some recent work [39, 33] on applications of affine subspace to vision problems. Wang *et al.* [39] propose an affine subspace based descriptor which can handle image transformation, including scaling, rotation and translation. Shirazi *et al.* [33], focusing on object tracking, model the object appearance variation by a set of affine subspaces and propose a measure based on Grassmann geodesic distance to compare the difference between two affine subspaces.

Building upon the assumption that data are drawn from a union of linear (or affine) subspaces, the subspace segmentation or clustering methods study how to estimate the number, dimensions, and basis of the subspaces. The statistical methods [35, 13] suppose that the random vector in each cluster follows Gaussian distribution and exploit the Expectation Maximization (EM) algorithm for subspace estimation. Some recent works present subspace segmentation approaches based on the sparse coding [7, 10] or low-rank representation [23]. A comprehensive review of the subspace segmentation methods are beyond our scope and the reader may refer to [8]. Note that our paper focus on a new feature encoding method with locality constraint given an ensemble of affine subspaces. We demonstrate the effectiveness of our idea with simple subspace segmentation method. By using state-of-the-art methods [46, 26, 11], the performance of LASC may be further improved. However, application of these methods to our problem may not be straightforward, as we often face a large number of high dimensional training data.

3. Locality-Constrained Linear Subspace Coding (LASC)

In this section, we first recall briefly the LLC method which is the starting point of our work (§ 3.1); after formulating the objective function of LASC (§ 3.2), we present the proximity measures (§ 3.3) and how to leverage the second-order information (§ 3.4).

3.1. LLC in Retrospect

Let $\mathbf{B} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M]$ be the dictionary consisting of M visual words $\boldsymbol{\mu}_i \in \mathbb{R}^n, i = 1, \ldots, M$, and $\mathbf{y}$ be an input feature to be encoded. The LLC is formulated as the following optimization problem:

$$\arg\min_{\mathbf{c}} \|\mathbf{y} - \mathbf{B}\mathbf{c}\|_2^2 + \lambda \sum\nolimits_i d(\mathbf{y}, \mathbf{b}_i) c_i^2 \quad s.t. \ \sum\nolimits_i c_i = 1. \tag{1}$$

Here $\lambda > 0$ is the regularization parameter, c_i is the i^{th} component of $\mathbf{c}$, and $d(\mathbf{y}, \mathbf{b}_i) = \exp(\beta \|\mathbf{y} - \mathbf{b}_i\|_2^2), \beta > 0$, where $\|\cdot\|_2$ indicates the Euclidean distance. By using the ℓ_2 regularization, LLC produces the sparse code which is locally smooth, a property that SC do not have. It is not difficult to see that there is a (unique) closed form solution to the problem (1). This enables LLC to be more efficient than SC which involves computationally demanding iterations.

The approximated version of LLC is to select directly the k-nearest words and represent $\mathbf{y}$ by a linear decomposition of them

$$\arg\min_{\mathbf{c}^{\mathcal{N}}} \|\mathbf{y} - \mathbf{B}^{\mathcal{N}} \mathbf{c}^{\mathcal{N}}\|_2^2, \ s.t. \ \sum\nolimits_i c_i^{\mathcal{N}} = 1, \tag{2}$$

where $\mathbf{B}^{\mathcal{N}}$ is the dictionary consisting of only k-nearest visual words to the input feature $\mathbf{y}$. This reduces considerably the computations as one only needs to solve a much smaller ($k \ll M$) system of linear equations. Note that throughout the paper [47] the approximated version of LLC (2) is adopted. As observed in [24], when employing the distances to all words the soft-assignment methods deteriorate, and they attributed this to that distances between the feature and remote words are not unreliable any more due to the underlying geometric structure of the manifold.

3.2. Formulation of LASC

Now we extend the idea of locality constraint from dictionary of visual words to dictionary of affine subspaces. In our method, the geometry of feature space is represented by an ensemble of low-dimensional subspaces attached to the representative points or affine subspaces:

$$\mathcal{S}_i = \left\{ \boldsymbol{\mu}_i + \mathbf{A}_i \mathbf{x}_i,\ \mathbf{x}_i \in \mathbb{R}^p \right\},\ i = 1, \ldots, M \qquad (3)$$

where $\boldsymbol{\mu}_i$ indicates the representative point and $\mathbf{A}_i$ is an $n \times p$ matrix whose columns form a basis of the linear subspace. Indeed, $\mathcal{S}_i$ defines a local coordinate system and all of them gives a holistic picture of the manifold.

Our idea is to represent a feature $\mathbf{y}$ by its top-k most neighboring affine subspaces, and in the meantime constraining the projection of $\mathbf{y}$ in each subspace by the proximity measure of the feature to this subspace. Specifically, the objective function of LASC is formulated as

$$\min_{\forall \mathbf{x}_i} \sum_{\mathcal{S}_i \in \mathcal{N}_k^S(\mathbf{y})} \left\{ \left\| (\mathbf{y} - \boldsymbol{\mu}_i) - \mathbf{A}_i \mathbf{x}_i \right\|_2^2 + \lambda d(\mathbf{y}, \mathcal{S}_i) \left\| \mathbf{x}_i \right\|_2^2 \right\}, \qquad (4)$$

where $\lambda > 0$ is a regularization parameter, $\mathcal{N}_k^S(\mathbf{y})$ is the neighborhood of $\mathbf{y}$ defined by the k closest subspaces, and $d(\mathbf{y}, \mathcal{S}_i)$ indicates the value of proximity measure of $\mathbf{y}$ to subspace $\mathcal{S}_i$. Here the proximity implies a measure which is to be discussed in § 3.3. It is clear that the objective function (4) decouples into independent Ridge regression problems in $\mathbf{x}_i$. The solution to (4) has closed form:

$$\mathbf{x}_i = \left(\mathbf{A}_i^T \mathbf{A}_i + \lambda d(\mathbf{y}, \mathcal{S}_i) \mathbf{I} \right)^{-1} \mathbf{A}_i^T (\mathbf{y} - \boldsymbol{\mu}_i) \qquad (5)$$

for $\mathcal{S}_i \in \mathcal{N}_k^S(\mathbf{y})$; and $\mathbf{x}_i$ is equal to zero otherwise, where $\mathbf{I}$ is the identity matrix of order p.

We segment the feature space by the k-means algorithm to obtain clusters $\mathcal{C}_i, i = 1, \ldots, M$. Let $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma_i}, i = 1, \ldots, M$ be the mean vector and covariance matrix of cluster $\mathcal{C}_i$. We assume, in the local coordinate system with the cluster center as origins, each cluster has a geometrical structure of low-dimensional linear subspace. We employ PCA to preserve the directions with larger variances. Let us denote by $\sigma_{i,1}^2, \ldots, \sigma_{i,n}^2$ the positive eigenvalues of $\boldsymbol{\Sigma}_i$ in non-decreasing order. Let $\mathbf{u}_{i,j}$ be the orthogonal eigenvector corresponding to $\sigma_{i,j}^2$. We employ the most significant principal directions $\mathbf{u}_{i,j}, j = 1, \ldots, p$ to form our PCA basis, i.e.,

$$\mathbf{A}_i = \mathbf{U}_i = \left[\mathbf{u}_{i,1}, \cdots, \mathbf{u}_{i,p} \right] \qquad (6)$$

In this case, the solution (5) has a simple form as follows:

$$\mathbf{x}_i = w_{\mathbf{y}}^i \mathbf{U}_i^T (\mathbf{y} - \boldsymbol{\mu}_i) \qquad (7)$$

for $\mathcal{S}_i \in \mathcal{N}_k^S(\mathbf{y})$, and $w_{\mathbf{y}}^i = (1 + \lambda d(\mathbf{y}, \mathcal{S}_i))^{-1}$. It can be seen that $\mathbf{x}_i$ is the orthogonal projection of $\mathbf{y} - \boldsymbol{\mu}_i$ in the subspace $\mathcal{S}_i$, weighted by the proximity measure-based weight $w_{\mathbf{y}}^i$. Thus far we can write out the *first-order* LASC vector for the feature $\mathbf{y}$

$$\mathbf{x} = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_i^T, \ldots, \mathbf{x}_M^T \right]^T \qquad (8)$$

where $\mathbf{x}_i$ is computed by Eq. (7) if the subspace $\mathcal{S}_i$ lies within $\mathcal{N}_k^S(\mathbf{y})$, and otherwise $\mathbf{x}_i$ is equal to zero vector.

3.3. Proximity Measure of Points to Affine Subspaces

In the LASC method we need to evaluate the degree of proximity measure of features to affine subspaces. In the following, we consider three kinds of proximity measures.

We first consider the proximity measure $d_r(\mathbf{y}, \mathcal{S}_i)$ from the traditional perspective of the reconstruction error. The minimum reconstruction error $\epsilon(\mathbf{y}, \mathcal{S}_i)$ of $\mathbf{y}$ in the affine subspace $\mathcal{S}_i$ is

$$\begin{aligned} \epsilon(\mathbf{y}, \mathcal{S}_i) &= \min_{\mathbf{x}_i} \left\| (\mathbf{y} - \boldsymbol{\mu}_i) - \mathbf{A}_i \mathbf{x}_i \right\|_2^2 \\ &= \left\| (\mathbf{I} - \mathbf{U}_i \mathbf{U}_i^T)(\mathbf{y} - \boldsymbol{\mu}_i) \right\|_2^2, \end{aligned} \qquad (9)$$

where T indicates the matrix transpose. Hence, we can naturally define

$$d_r(\mathbf{y}, \mathcal{S}_i) = \exp(\beta \epsilon^2(\mathbf{y}, \mathcal{S}_i)). \qquad (10)$$

The parameter β is obtained by the cross-validation method [15].

Next, we introduce the proximity measure from the statistical perspective. Let us consider a simple but more general case. We assume that the probability distribution $p(\mathbf{y}|\mathcal{C}_i)$ of all clusters $\mathcal{C}_i, i = 1, \ldots, M$ are isotropic Gaussians, whose mean vectors are their respective cluster centers and whose covariance matrices are $\sigma^2 \mathbf{I}$. We further assume that the prior probability $p(\mathcal{S}_i), i = 1, \ldots, M$ are uniform, then in terms of the Bayes' rule, we have

$$d_s(\mathbf{y}, \mathcal{S}_i) = \frac{\exp(-\frac{1}{2\sigma^2} \|\mathbf{y} - \boldsymbol{\mu}_i\|_2^2)}{\sum_{i'=1}^{M} \exp(-\frac{1}{2\sigma^2} \|\mathbf{y} - \boldsymbol{\mu}_{i'}\|_2^2)}. \qquad (11)$$

In this case, the proximity measure of one point to the affine subspace reduces to the Euclidean space between this point to the cluster center $\boldsymbol{\mu}_i$. The underlying assumption is that we approximate the feature distribution by an ensemble of spheres of the same radius. This method has very loose assumption which may be scalable to unknown data. We also use the cross validation technique to determine the value of σ. Note that this kind of statistical modeling method has been used in image classification [36, 24].

Finally, we derive the proximity measure from training data, also from the statistical perspective. As $\mathbf{y}$ is in the low-dimensional subspace and the random vector

$$\mathbf{z}_i = \mathbf{U}_i^T (\mathbf{y} - \boldsymbol{\mu}_i), \qquad (12)$$

is the orthogonal projection of $\mathbf{y}$ in the affine space $\mathcal{S}_i$. We denote by $z_{i,j} = \mathbf{u}_{i,j}^T (\mathbf{y} - \boldsymbol{\mu}_i)$ the j^{th} component of $\mathbf{z}_i$. From PCA we know that the expectation and covariance matrix of $\mathbf{z}_i$ are

$$\mathrm{E}(\mathbf{z}_i) = \mathbf{0}, \ \ \mathrm{cov}(\mathbf{z}_i) = \mathrm{diag}(\sigma_{i,1}, \ldots, \sigma_{i,p}), \qquad (13)$$

respectively, where $\mathrm{diag}(\sigma_{i,1}, \ldots, \sigma_{i,p})$ is the diagonal matrix. Different components $z_{i,j}$ in the random vector $\mathbf{z}_i$ are not correlated. Here, we assume that $\mathbf{z}_i$ follows Gaussian distribution with mean vector $\mathrm{E}(\mathbf{z}_i)$ and covariance matrix $\mathrm{cov}(\mathbf{z}_i)$, i.e.,

$$p(\mathbf{z}_i|\mathcal{S}_i) = \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi}\sigma_{i,j}} \exp(-\frac{z_{i,j}^2}{2\sigma_{i,j}^2}). \tag{14}$$

We define

$$\begin{aligned} d_p(\mathbf{y}, \mathcal{S}_i) &= p(\mathcal{S}_i|\mathbf{z}_i) \qquad (15) \\ &= \frac{\prod_{j=1}^{p} \frac{1}{\sigma_{i,j}} \exp(-\frac{z_{i,j}^2}{2\sigma_{i,j}^2})}{\sum_{i=1}^{M} \prod_{j=1}^{p} \frac{1}{\sigma_{i,j}} \exp(-\frac{z_{i,j}^2}{2\sigma_{i,j}^2})}. \end{aligned}$$

We derive (15) again based on the Bayes' rule and assume the prior probability $p(\mathcal{S}_i), i = 1, \ldots, M$ be uniform. Note that in this case the $\mathrm{cov}(\mathbf{y}) = \mathrm{E}\{(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})^T\}$ has rank $p < n$, which follows the so-called *degenerate* Gaussian distribution [32].

3.4. Combination of the Second-Order Information

The second-order information has proven helpful for increasing classification accuracy [31, 4]. Due to success of the Fisher vector method [31], in the following we propose our second-order LASC based in Fisher information metric as well. It is known [17] that the space of exponential family of distributions, e.g., Gaussian distribution, forms a statistical manifold on which the Riemannian metric is defined by the Fisher information metric. The Fisher vector is the gradient of the likelihood function with respect the parameters of the distributions normalized by the Fisher information matrix. Indeed, it is the gradient on the tangent space of the statistical manifold (called natural gradient) [1].

Let us consider the Fisher vector associated with the random vector $\mathbf{z}_i$ which has probability density function (PDF) described by (14). For notational clarity, below we use $p(\mathbf{z}_i|\boldsymbol{\lambda}_i)$ to denote the PDF of $\mathbf{z}_i$, where $\boldsymbol{\lambda}_i = [\sigma_{i,1} \ \ldots \ \sigma_{i,p}]^T$ is the parameter vector. Let $\mathbf{g}_{\lambda_i} = \triangledown_{\lambda_i} \log(p(\mathbf{z}_i))$ be the gradient vector of the likelihood function $\log(p(\mathbf{z}_i))$. We can derive the Fisher information matrix $\mathbf{F}_{\boldsymbol{\lambda}_i} = \mathbf{E}_{p(\mathbf{z}_i|\lambda_i)}(\mathbf{g}_{\lambda_i}\mathbf{g}_{\lambda}^T) = \mathrm{diag}(\sigma_{i,1}^{-2} \ \ldots \ \sigma_{i,p}^{-2})$, where $\mathbf{E}_{p(\mathbf{z}_i|\lambda_i)}(\cdot)$ indicates the expectation with respective to $p(\mathbf{z}_i|\lambda_i)$. The Fisher vector is computed by $\mathbf{f}_{\boldsymbol{\lambda}_i} = \mathbf{F}_{\boldsymbol{\lambda}_i}^{-1/2}\mathbf{g}_{\lambda_i}$, where $\mathbf{F}_{\boldsymbol{\lambda}_i}^{-1/2}$ is the inverse of the square root matrix of $\mathbf{F}_{\boldsymbol{\lambda}_i}$. After some derivation, we can obtain the Fisher vector and accordingly, define our second-order LASC vector as

$$\mathbf{x}_i^{.2} = w_{\mathbf{y}}^i \mathbf{f}_{\lambda_i} = \frac{w_{\mathbf{y}}^i}{\sqrt{2}} \left[\frac{z_{i,1}^2}{\sigma_i^2} - 1, \ldots, \frac{z_{i,p}^2}{\sigma_p^2} - 1 \right]^T. \tag{16}$$

The final LASC vector, containing both the first-order and second-order information, is written as

$$\mathbf{x} = \begin{bmatrix} \vdots \\ \mathbf{x}_i \\ \mathbf{x}_i^{.2} \\ \vdots \end{bmatrix}, \overbrace{\underbrace{\mathbf{x}_i = w_{\mathbf{y}}^i \begin{bmatrix} z_{i,1} \\ \vdots \\ z_{i,p} \end{bmatrix}}_{1^{\mathbf{st}}\text{-order vector}}, \underbrace{\mathbf{x}_i^{.2} = \frac{w_{\mathbf{y}}^i}{\sqrt{2}} \begin{bmatrix} \left(\frac{z_{i,1}}{\sigma_{i,1}}\right)^2 - 1 \\ \vdots \\ \left(\frac{z_{i,p}}{\sigma_{i,p}}\right)^2 - 1 \end{bmatrix}}_{2^{\mathbf{nd}}\text{-order vector}}}^{\text{LASC vector if } \mathcal{S}_i \in \mathcal{N}_k^S(\mathbf{y})\text{; otherwise } \mathbf{x}_i = \mathbf{x}_i^{.2} = \mathbf{0}} \tag{17}$$

4. Experiments

In this section, we make experiments on four benchmark datasets to test the performance of the proposed LASC. First, we introduce the experimental setting and the benchmarks. We then assess the important components in LASC on the challenging PASCAL VOC 2007, which include the number of subspaces, subspace dimension, proximity measures, and utilization of the second-order information. Finally, we compare with the state-of-the-art methods.

4.1. Experimental Setting

In all our experiments, we extract five scale dense SIFT features with stride of four pixels. We learn dictionaries of affine subspaces by using the k-means algorithm with over five million features. Three level spatial pyramid [22] (1×1, 3×1 and 2×2) are used in the proposed LASC. For the image-level LASC vector, we normalize the first-order and second-order sub-vectors separately per subspace by l_2 norm, and then l_2-normalize the aggregated vector on each sub-image in the pyramid. The SVM classifiers are learned in a one-vs-all fashion to deal with multi-class classification problem. All the programs are written in Matlab release 2013a running on a PC with Intel(R) Core(TM) i7-4820K CPU @ 3.40GHz and 64G RAM. We implement extraction of SIFT and SVM classifier with VLFeat [37].

4.1.1 Benchmark datasets

We employ four challenging benchmark datasets for performance evaluation, where two datasets are for the object class recognition and the other two for scene categorization.

PASCAL VOC 2007 [9] is a very challenging benchmark which contains 9,963 images from 20 object categories with large within-class variations. We follow the standard evaluation protocol [9] and report the mean average precision (mAP).

Caltech 256 [12] is composed of 30,607 images in 256 object categories and one background class. It contains large variations of the object size and pose. Following the custom setup, we test the LASC algorithm by randomly selecting 15, 30, 45, and 60 training images per class and all the rest for testing, background class is not evaluated. The experiments are repeated five times and the average accuracy and standard deviation are reported.

MIT Indoor 67 [29] is a difficult indoor scene dataset due to the large variability of within-class and large confusion between-class. It contains 67 categories, each of which has at least 100 images and 15,620 images in total. We use a subset of the dataset together with fixed training/test splits as in [29] and report the mean accuracy.

SUN 397 [40] is a large database for scene categorization. It contains more than 100K well-sampled images from 397 indoor and outdoor scene categories. Each category has 100 images at least. Following the experimental setting in [40], we use ten chosen subsets for evaluation. In each subset, 5, 10, 20 or 50 samples per class are used for training and 50 samples per class for testing. The average accuracy of ten subsets is reported.

4.2. Analysis of LASC

In this subsection, we are to conduct a sequence of experiments on the VOC 2007 [9] to analyze the influence of the important parameters on the LASC. The parameter β in the proximity measure d_r (10) and σ in the proximity measure d_s (11) are set to 1500 and 0.1, respectively, by the cross-validation method. The regularization parameter $\lambda = 1$ in Eq.(4) throughout the paper. The parameters in the proximity measure d_p (15) are determined from training samples as described in Section 3.

4.2.1 Number of nearest subspaces

In the LASC method, we need first to decide the number k of nearest subspaces. Here we evaluate how k affects the performance of the first-order LASC. The number M of subspaces is set to 256 and dimension p of each subspace is set to 64. We choose d_s to produce only the first-order LASC vectors as d_s is simple yet more general in the sense that it does not involve the subspace structure. Figure 2 shows the curve of the mAP of LASC vs. number of nearest subspaces. It can be seen that the mAP reaches peak at $k = 3$ and then smoothly decreases as k gets larger. We mention that d_r and d_p also achieve the best results when $k = 3$, but when $k > 3$, their performances decrease less dramatically than that of d_s. We set $k = 3$ in the remaining experiments.

4.2.2 Subspace dimension

The LASC method assumes that the data manifold can be effectively represented by an ensemble of subspaces attached to the representative points. Below we are to validate this assumption, and study the influence of subspace dimension. For this purpose, we set $M = 256$, and use the proximity measure d_s to produce only the first-order LASC vectors. We plot the curve of mAP vs. subspace dimension p (the number of principal components in our case) in Figure 3. The classification performance of LASC increases rapidly with dimension until $p = 64$. Increasing dimension from 64 to 128 achieves no more than 0.26% mAP growth. It can be seen that too small dimensions are insufficient to describe the structure of the subspace, while much larger ones give little benefit. The behaviors of d_r and d_p are very similar to that of d_s. The results confirm our hypothesis that the feature data lie in certain low-dimensional subspaces. We set $p = 64$ in all the subsequent experiments to counterbalance the efficiency and performance.

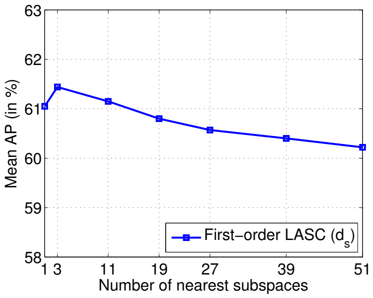

Figure 2. Influence of number of nearest subspaces on the LASC method on VOC 2007.

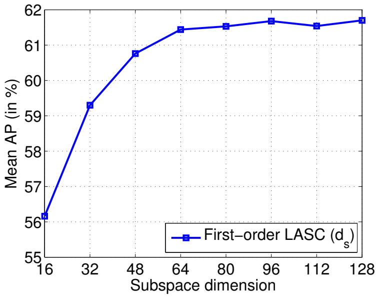

Figure 3. Influence of subspace dimensions on the LASC method on VOC 2007.

4.2.3 Proximity measures and comparison with LLC

The proximity measures between descriptors and affine subspaces play a crucial role in proposed LASC method. We hereby evaluate the proximity measures d_r (10), d_s (11) and d_p (15). We also only use the first-order LASC vector. The mAPs of the three methods vs. dictionary size M are shown in Figure 4. We can see that LASC (d_s) and LASC (d_p) outperform LASC (d_r) by a large margin while LASC (d_p) is coherently better than LASC (d_s). Interestingly, the margin between LASC (d_s) and LASC (d_p) gradually gets smaller with increasing dictionary size until is negligible at $M = 512$. This is a desirable property as described in Section 3.3, in the case of d_s we can use Euclidean distance to find the k-nearest subspaces which makes the proposed method very efficient. For all the following experiments, we select the proximity measure d_p.

To analyze why various proximity measures perform differently, we randomly select 50 features from the set of training features, and compute the values of proximity measures of these features to all the affine subspaces. The results are displayed in Figure 5. The values of d_p (top) appear very sparse in the sense that for a feature few proximity measure values are significant while others being negligible, which indicates that the nearest subspaces can be found accurately. In sharp contrast, the values of d_r (middle) are densely populated, indicating that the nearest subspaces can not be determined reliably. The case of d_s (bottom) is intermediate between those of d_p and d_r.

Currently LASC (d_p) is computationally demanding due to calculation of posterior probability of features (CPPF), which is implemented in Matlab language without any optimization. We can implement CPPF in C language and optimize the codes to improve its efficiency, as done in [37]. LASC (d_r) takes comparable time with LASC (d_s) while LASC (d_p) is very efficient as only Euclidean distances are involved.

Comparison with LLC. We also compare the first-order LASC method with LLC [38]. To make the comparison as fair as possible, for LASC with M subspaces of d-dimension, we use the dictionary of Md visual words for LLC such that both have coding vector of same size. As shown in Figure 4, LASC (d_s) and LASC (d_p) have clear advantages over LLC, particularly as the dictionary size gets larger. We owe the higher performance of LASC to the usage of encoding on affine subspaces. LLC provides only piecewise constant approximation of the data manifold and enlarging the number of words in the dictionary helps but fails to address the problem. In contrast, the first-order LASC has better representation capability by providing the piecewise linear approximation of the manifold.

4.2.4 Combination of second-order information

The purpose of experiments in this section is to evaluate the performance of LASC by using separately the first- and second-order information as well as by combining them. Figure 6 presents the mAP of the respective methods vs. dictionary size. The second-order LASC has similar performance with the first-order one with smaller dictionary size ($M < 32$), but the latter one outperforms by increasingly large margins. By combining the first- and second-order information, LASC (1st+2nd) achieves significant performance boost.

The FV [31] method also leverages the first- and second-order information and we compare it with the proposed LASC. Note that the LASC method using only the first-order information are better than FV when $M > 256$. The LASC (1st+2nd) coherently outperforms FV by average 1.5% mAP ($M \geq 64$) and the highest gains is 2.0% ($M = 512$).

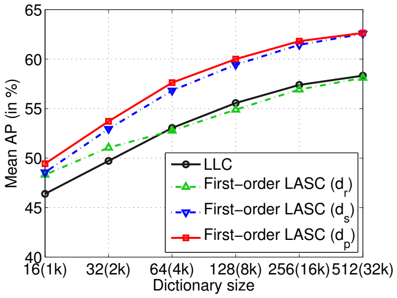

Figure 4. Comparison of LASC (first-order) by using different proximity measures and comparison with LLC on VOC 2007. The numbers in the brackets on the horizontal axis indicate the dictionary size of LLC.

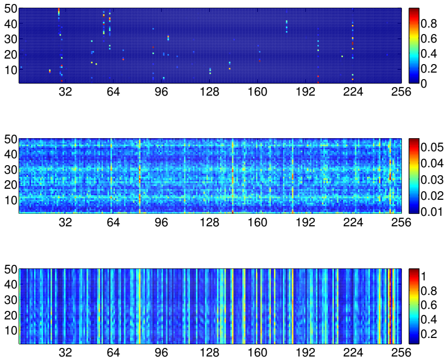

Figure 5. Proximity measures d_p (top), d_r (middle) and d_s (bottom) of features (vertical axis) to all affine subspaces (horizontal axis) in the dictionary of LASC.

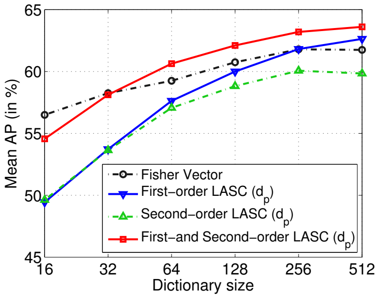

Figure 6. Comparison of the LASC methods by using the first-order or/and second-order information on VOC 2007. The results of FV are also shown for comparison.

4.3. Comparison with state-of-the-art methods

PASCAL VOC 2007 [9]. In Section 4.2.4, we have compared the performance of FV and LASC against varying dictionary size. Table 1 summarizes comparison results with other methods besides FV. The result (63.6%) of LASC achieved with 512 subspaces outperforms LLC by a large margin (6%), and is much better than SV [47] as well as the PASCAL VOC 2007 winner [9]. Our result is also comparable with the state-of-the-art results, i.e., 63.5% in [14] and 63.8% in [21]. Note that Harzallah *et al.* [14] combined the localization (detection) and classification modules and Kobayashi [21] introduced an improved FV method by using Dirichlet-based GMM.

Caltech 256 [12]. We compare LASC with six state-of-the-art methods [42, 38, 47, 31, 2, 21], and the results are summarized in Table 2. On this benchmark, the proposed LASC method outperforms all these six methods by using any number of training samples. Particularly, as regards the average classification accuracy over four kinds of training samples, the LASC method (1) outperforms FV by about 5.1%; (2) achieves approximately 2.4% higher classification accuracy than Kobayashi [21] which concerns an improved FV method; and (3) has an advantage of 1.7% higher accuracy over Bo *et al.* [2], which trained a three layer deep architecture for learning image representation.

MIT Indoor 67 [29]. Table 3 presents the comparison results of LASC and the competing methods. The proposed LASC achieves 1.6% higher accuracy than FV, and has much better performance than other methods except [41], which achieves the highest classification accuracy, slightly higher than ours by 0.1%. Their method is specially designed for indoor scene recognition by combining spatial pyramid matching (SPM) and orientational pyramid matching (OPM) based on Fisher vector [31].

SUN 397 [40]. We present in Table 4 the comparison results of LASC with other methods, where the results of LLC and SV are reproduced from [16]. The LASC evidently has much higher classification accuracy than Xiao *et al.* [40], LLC (4k) [38] and SV [47]. It also coherently outperforms the FV method [31], and the performance gap gets larger as the number of training samples increases (the largest gap is 2.0% with 50 training samples). Note that Kobayashi [21] reported 46.1% accuracy with 50 training samples, which is slightly higher than ours.

Similar to LASC, the FV method also exploits Fisher information metric and performs local coding, i.e., coding of one feature with respect to 5∼10 Gaussians with significant posterior probabilities [31, Appendix 2]. Despite these similarities, FV and LASC have big differences. The FV method uses a global orthogonal basis obtained by PCA for dimensionality reduction plus (also in that global system) training of a universal GMM for modeling feature statistics. In a sharp contrast, the LASC method leverages an ensemble of local coordinate systems of varying origins and the corresponding local bases; the dimensionality reduction and the coding are both relative to the local bases. In our opinion, the ensemble of local bases can better represent the geometry of data manifold, which distinguishes the proposed LASC from most of the existing coding methods, and may account for its superior performance.

Method	mAP (%)
LLC (25k) [5]	57.6
SV (1k) [5]	58.2
The winners [9]	59.4
FV (256) [31]	61.8
Harzallah *et al.* [14]	63.5
Kobayashi [21]	**63.8**
LASC (256)	63.2
LASC (512)	63.6

Table 1. Comparison on Pascal VOC 2007.

# samples	15	30	45	60
SC (1k) [42]	27.7 (0.5)	34.0 (0.4)	37.5 (0.6)	40.1 (0.9)
LLC (4k) [38]	34.4 (-)	41.2 (-)	45.3 (-)	47.7 (-)
SV (256) [16]	36.1 (-)	42.4 (-)	46.3 (-)	48.8 (-)
FV (256) [31]	38.5 (0.2)	47.4 (0.1)	52.1 (0.4)	54.8 (0.4)
Kobayashi [21]	41.8 (0.2)	49.8 (0.1)	54.4 (0.3)	57.4 (0.4)
Bo *et al.* [2]	42.7 (-)	50.7 (-)	54.8 (-)	58.0 (-)
LASC (256)	**43.7 (0.4)**	**52.1 (0.1)**	**57.2 (0.3)**	**60.1 (0.3)**

Table 2. Comparison on Caltech 256.

Method	Acc. (%)
Quattoni *et al.* [29]	26.0
SV (1k) [20]	56.2
FV (256) [31]	61.3
Bo *et al.* [2]	51.2
Kobayashi [21]	63.4
Xie *et al.* [41]	**63.5**
LASC (256)	62.9
LASC (512)	63.4

Table 3. Comparison on MIT Indoor 67.

# samples	5	10	20	50
Xiao *et al.* [40]	14.5	20.9	28.1	38.0
LLC (4k) [16]	13.5	18.7	24.5	32.4
SV (128) [16]	16.4	21.9	28.4	36.6
FV (256) [31]	19.2 (0.4)	26.6 (0.4)	34.2 (0.3)	43.3 (0.2)
LASC (256)	**19.4 (0.4)**	**27.3 (0.3)**	**35.6 (0.1)**	**45.3 (0.4)**

Table 4. Comparison on SUN 397.

5. Conclusion

This paper presented a novel method called LASC for feature encoding, which extended the locality-constrained linear coding (LLC) from an ensemble of visual words to a dictionary of affine subspaces. In the LASC method, the manifold of high dimensional features is described by local subspaces attached to the representative points. This fundamentally differs from LLC which describes the data manifold only by the representative points (i.e., visual words), while ignoring the geometric structure immediately surrounding them. We also propose to exploit the second-order information in each subspace based on the Fisher information metric. In the future, it is interesting to integrate into LASC the histogram feature transform method [21] which has greatly improved FV. Advanced methods for subspace segmentation rather than the simple k-means clustering may further benefit the LASC method.

References

[1] S. Amari. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, 1998. 5

[2] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *CVPR*, 2013. 8

[3] G. D. Canas, T. Poggio, and L. Rosasco. Learning manifolds with k-means and k-flats. *NIPS*, 2012. 1, 2

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic Segmentation with Second-Order Pooling. In *ECCV*, 2012. 3, 5

[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *BMVC*, 2011. 3, 8

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004. 1

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 3

[8] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 35(11):2765–2781, 2013. 3

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010. 5, 6, 7, 8

[10] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR*, 2011. 3

[11] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, 2014. 3

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 5, 8

[13] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *CVPR*, 2004. 3

[14] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 8

[15] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009. 2, 3, 4

[16] Y. Huang, Z. Wu, L. Wang, and T. Tan. Feature coding in image classification: A comprehensive study. *TPAMI*, 36(3):493–506, 2014. 1, 2, 3, 8

[17] S. ichi Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 2000. 3, 5

[18] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 1

[19] N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Comput.*, 9(7):1493–1516, 1997. 2, 3

[20] T. Kobayashi. BoF meets HOG: Feature extraction based on histograms of oriented p.d.f gradients for image classification. In *CVPR*, 2013. 8

[21] T. Kobayashi. Dirichlet-based histogram feature transform for image classification. In *CVPR*, 2014. 3, 8

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 5

[23] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010. 3

[24] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, 2011. 1, 3, 4

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[26] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin. Correntropy induced l2 graph for robust subspace clustering. In *ICCV*, 2013. 3

[27] H. Nakayama. Aggregating descriptors with local gaussian metrics. In *NIPS Workshop*, 2012. 3

[28] X. Peng, L. Wang, Y. Qiao, and Q. Peng. Boosting VLAD with supervised dictionary learning and high-order statistics. In *ECCV*, pages 660–674, 2014. 3

[29] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 6, 8

[30] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1, 2

[31] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *IJCV*, 105(3):222–245, 2013. 1, 3, 5, 7, 8

[32] V. Schmidt. Stochastics III. Technical report, Tilburg University, 2012. 5

[33] S. A. Shirazi, C. Sanderson, C. McCool, and M. T. Harandi. Improved object tracking via bags of affine subspaces. *CoRR*, 2014. 3

[34] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1

[35] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Comput.*, 11(2):443–482, 1999. 3

[36] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *TPAMI*, 32(7):1271–1283, 2010. 1, 4

[37] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 5, 7

[38] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 1, 7, 8

[39] Z. Wang, B. Fan, and F. Wu. Affine subspace representation for feature description. In *ECCV*, 2014. 3

[40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6, 8

[41] L. Xie, J. Wang, B. Guo, B. Zhang, and Q. Tian. Orientational pyramid matching for recognizing indoor scenes. In *CVPR*, 2014. 8

[42] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1, 8

[43] J. Yang, K. Yu, and T. S. Huang. Efficient highly overcomplete sparse coding using a mixture model. In *ECCV*, 2010. 2

[44] K. Yu and T. Zhang. Improved local coordinate coding using local tangents. In *ICML*, 2010. 1, 2

[45] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *NIPS*, 2009. 1, 2

[46] Y. Zhang, Z. Sun, R. He, and T. Tan. Robust subspace clustering via half-quadratic minimization. In *ICCV*, 2013. 3

[47] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010. 1, 2, 3, 7, 8