Two-Layers Local Coordinate Coding

Wei Xiao[1](✉), Hong Liu[1], Hao Tang[1], and Huaping Liu[2]

[1] Engineering Lab on Intelligent Perception for Internet of Things (ELIP),
Key Laboratory for Machine Perception, Shenzhen Graduate School,
Peking University, Beijing, China
xiaoweithu@163.com

[2] State Key Laboratory of Intelligent Technology and Systems,
Department of Computer Science and Technology,
Tsinghua University, Beijing, China

Abstract. Extracting informative regularized representations of input signals plays a key role in the field of artificial intelligence, such as machine learning and robotics. Traditional approaches feature ℓ_2 norm and sparse inducing ℓ_p norm ($0 \leq p \leq 1$) based optimization methods, imposing strict regularization on the representations. However, these approaches overlook the fact that signals and atoms in the overcomplete dictionaries usually contain such wealth of structural information that could improves representations. This paper systematically exploits data manifold geometric structure where signals and atoms reside in, and thus presents a principled extension of sparse coding, i.e. two-layers local coordinate coding, which demonstrates a high dimensional nonlinear function could be locally approximated by a global linear function with quadratic approximation power. Moreover, to learn each latent layer, corresponding patterned optimization approaches are developed, encoding distance information between signals and atoms into the representations. Experimental results demonstrate the significance of this extension on improving the image classification performance and its potential applications for object recognition in robot system are also exploited.

Keywords: Local coordinate coding · Machine learning · Sparse coding · Robotics

1 Introduction

Recent years have witnessed a fast growing interest in the research on sparse representations of signals with overcomplete dictionaries. Scholars from various research fields promote the progress, such as, Donoho from the statistics community [1], Elad from the machine learning community[2], and Kouskouridas from the robotics community[3], etc. Recent theoretical analyses observed that sparse

This work is supported by National Natural Science Foundation of China (NSFC, No. 61340046), National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247), Scientific and Technical Innovation Commission of Shenzhen Municipality (No. JCYJ20130331144631730).

H. Zha et al. (Eds.): CCCV 2015, Part I, CCIS 546, pp. 34–45, 2015.
DOI: 10.1007/978-3-662-48558-3_4

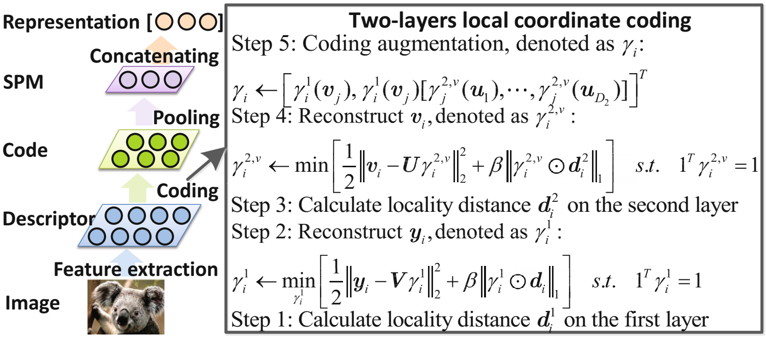

Representation [○○○]
Concatenating
SPM
Pooling
Code
Coding
Descriptor
Feature extraction
Image
Two-layers local coordinate coding
Step 5: Coding augmentation, denoted as γ_i:
$\gamma_i \leftarrow \left[\gamma_i^1(\boldsymbol{v}_j), \gamma_i^1(\boldsymbol{v}_j)[\gamma_j^{2,v}(\boldsymbol{u}_1), \cdots, \gamma_j^{2,v}(\boldsymbol{u}_{D_2})]\right]^T$
Step 4: Reconstruct $\boldsymbol{v}_i$, denoted as $\gamma_i^{2,v}$:
$\gamma_i^{2,v} \leftarrow \min\left[\frac{1}{2}\|\boldsymbol{v}_i - \boldsymbol{U}\gamma_i^{2,v}\|_2^2 + \beta\|\gamma_i^{2,v} \odot \boldsymbol{d}_i^2\|_1\right] \quad s.t. \quad 1^T\gamma_i^{2,v} = 1$
Step 3: Calculate locality distance $\boldsymbol{d}_i^2$ on the second layer
Step 2: Reconstruct $\boldsymbol{y}_i$, denoted as γ_i^1:
$\gamma_i^1 \leftarrow \min_{\gamma_i^1}\left[\frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{V}\gamma_i^1\|_2^2 + \beta\|\gamma_i^1 \odot \boldsymbol{d}_i\|_1\right] \quad s.t. \quad 1^T\gamma_i^1 = 1$
Step 1: Calculate locality distance $\boldsymbol{d}_i^1$ on the first layer

Fig. 1. Left: The traditional image representation pipeline. Right: the proposed two-layers local coordinate coding.

representation could be extended to "local" representation: nonzero coefficients are often assigned to atoms nearby to the encoded point [4–6]. An extension to sparse representation, called Local Coordinate Coding (LCC) is thus proposed, which learns a nonliear function in high dimension by forming a set of local bases on the data manifold. The nonlinear function approximation view of sparse representation not only brings about in-depth understanding of its fundamental connotation and success, but also provides opportunities to get a deeper insight into its parentage ties with the essence – locality.

This paper follows these lines of research, and scratches the surface of its utilization potential in computer vision and robotics, where we try to make a principled extension of the traditional single-layer coding to a more generalized two-layers local representation problem, called Two-layers Local Coordinate Coding (Two-layers LCC). This coding strategy takes advantage of the underlying data manifold geometric structure to locally embed points on the manifold into a lower dimensional two-layers structure, see Figure 1. Therefore, Two-layers LCC turns a very difficult high dimensional nonlinear learning problem into a simpler linear learning problem, which could be effectively solved using for instance, ℓ_1 optimization. More important, it could achieve higher approximation power than its single-layer counterpart, especially in the situation of fewer or noise-polluted training samples, see theoretical analysis in Section 3.

The remainder of this paper is organized as follows: Section 2 surveys the evolution of related coding strategies. And Section 3 makes a theoretical introduction into the Two-layers LCC. Accordingly, specific coding formulations to each layer are proposed in Section 4. Experimental evaluations on popular benchmarks and a practical application in robotics are presented in Section 5 and conclusions are drawn in Section 6.

2 Prior Art

This section provides a brief review to help comprehend the underlying relationship between sparsity and locality.

One of popular extensions of sparse coding is LLC [5,7], which supposes that although signals $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N] = \{\boldsymbol{y}_i\}_{i=1}^N$, $\boldsymbol{y}_i \in \mathbb{R}^m$ are physically represented by $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] = \{\boldsymbol{x}_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathbb{R}^p$, where $p \gg m$, they often lie on a manifold with a much smaller intrinsic dimensionality. Specifically, let $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_p] \in \mathbb{R}^{m \times p}$ be a dictionary with p atoms in $\mathbb{R}^m$, for signals $\boldsymbol{Y}$, the corresponding LLC representations $\boldsymbol{X}$, can be obtained by solving:

$$\min_{\boldsymbol{x}} [\sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{x}_i\|_2^2 + \lambda \|\boldsymbol{b}_i \odot \boldsymbol{x}_i\|_2^2] \quad s.t. \quad \mathbf{1}^\mathsf{T}\boldsymbol{x}_i = 1, \tag{1}$$

where $\odot$ denotes the element-wise multiplication, and $\boldsymbol{b}_i \in \mathbb{R}^p$ is the locality adaptor that gives different freedom for each basis vector $\boldsymbol{d}_j$ proportional to its similarity of the input descriptor $\boldsymbol{y}_i$. Specifically,

$$\boldsymbol{b}_i = \exp(\frac{dist(\boldsymbol{y}_i, \boldsymbol{D})}{\sigma}), \tag{2}$$

where $dist(\boldsymbol{y}_i, \boldsymbol{D}) = [dist(\boldsymbol{y}_i, \boldsymbol{d}_1), \cdots, dist(\boldsymbol{y}_i, \boldsymbol{d}_p)]^\mathsf{T}$, and $dist(.,.)$ is the Euclidean distance, and σ is used for adjusting weight decay speed for locality.

Diverse representation strategies mentioned above can essentially be interpreted as taking fully advantages of infinitely many possible solutions $\boldsymbol{x}$ to the underdetermined systems of equation $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$ with different regularization terms to finally find a solo solution with desired suitable form. Compared with sparse coding, local coding can achieve: (i) more accurate correlations capturing and (ii) local smooth sparsity. For instance, the LLC can catch atoms structure of manifold where the signals reside in, and further use these atoms for coding; while sparse coding only pursues the solo goal, as sparse as possible in the final representation, as in the extreme case of sparse coding, i.e. vector quantization, only a few of atoms without structure would be selected. Similar to sparse coding, the LLC has achieve less reconstruction error by using multiple atoms [7], overcoming the shortcoming that sparse-inducing regularization terms are not smooth, thus provide incoherent atoms for similar signals to favor sparsity, losing correlations between codes.

However, LLC has a major disadvantage. In order to achieve higher approximation, one has to use a large number of so-called "anchor points", i.e. atoms close to the signal, to describe these signals. Finding enough powerful anchor points plays a key role in the representation pipeline. Unfortunately these anchor points are vulnerable to noise and inadequate training samples, and some of them are not necessarily have powerful descriptive ability. Therefore, it is eager to equip them with more descriptive power for better approximating $\boldsymbol{y}_i$ in order to guarantee accurate inferences from it. Shall we fix it or frankly speaking, fully explore the potential of the manifold to empower the anchors to better describe $\boldsymbol{y}_i$? The following section will give the answer.

3 Two-Layers Local Coordinate Coding

Let's first consider the problem of learning a nonlinear function $f(\boldsymbol{y})$ defined on a high dimensional space: $\mathbb{R}^m$, with large m. We have sampled this underlying distribution and obtained a set of labeled data: $(\boldsymbol{y}_1, l_1), \cdots, (\boldsymbol{y}_n, l_n)$. There are a lot of approaches to learn such a function in low dimension, while many of them are more or less suffer from the so-called "curse of dimensionality". One of intuitive explanation is since we do not have enough expressive data, i.e., $m \gg n$, it is hard to fully describe how this nonlinear function would appear in $\mathbb{R}^m$. One would argue, if we obtain more data points than n? While it is still hard due to the redundance of the data. However, the good news is, in many real applications with high dimensionality, we do not observe this so-called curse of dimensionality, this is because although data are physically represented in a high dimensional space, they often lie on a manifold which has a much smaller intrinsic dimensionality [4,6]. That is, many areas of this space are empty, or viewed empty.

The recent coding approach called LCC in [4] addresses this issue, which turns a difficult high dimensional nonlinear learning problem into a linear learning problem. While, its approximation accuracy is vulnerable to the limited anchor points, where this paper systematically makes effort to equip them with more descriptive power.

3.1 Lipschitz Smoothness

This section reviews the Lipschitz smoothness for further analysis of two-layers representation.

Definition 1 (Lipschitz Smoothness). *A function $f(\boldsymbol{y})$ on $\mathbb{R}^m$ is $(\alpha, \beta, \upsilon)$ Lipschitz smoothness with respect to a norm $\|\cdot\|$, if*

$$\begin{cases} |f(\boldsymbol{y}') - f(\boldsymbol{y})| \leq \alpha \|\boldsymbol{y}' - \boldsymbol{y}\|, & (3) \\ |f(\boldsymbol{y}') - f(\boldsymbol{y}) - \nabla f(\boldsymbol{y})^T(\boldsymbol{y}' - \boldsymbol{y})| \leq \beta \|\boldsymbol{y}' - \boldsymbol{y}\|^2, & (4) \\ |f(\boldsymbol{y}') - f(\boldsymbol{y}) - 0.5(\nabla f(\boldsymbol{y})^T + \nabla f(\boldsymbol{y}')^T)(\boldsymbol{y}' - \boldsymbol{y})| \leq \upsilon \|\boldsymbol{y}' - \boldsymbol{y}\|^3, & (5) \end{cases}$$

where we assume $\alpha, \beta, \upsilon \geq 0$, and the norm always refers to the Euclidean norm (ℓ_2 norm). These three types of smoothness would be used in the following derivations.

Lipschitz smoothness characterizes different levels of smoothness of function $f(\boldsymbol{y})$. Intuitively, Lipschitz smoothness offers an opportunity to zoom in on the function $f(\boldsymbol{y})$ at different levels, that is at $0th$ order level (constant approximation level), $f(\boldsymbol{y})$ could be roughly approximated by $f(\boldsymbol{y}')$, corresponding approximation quality could be measured by $\alpha \|\boldsymbol{y}' - \boldsymbol{y}\|$; at $1st$ order level (linear approximation level), $f(\boldsymbol{y})$ could be roughly approximated by $f(\boldsymbol{y}')$ and its gradient $\nabla f(\boldsymbol{y})^{\mathsf{T}}$, corresponding approximation quality could be measured by $\beta \|\boldsymbol{y}' - \boldsymbol{y}\|^2$; at $2nd$ order level (quadratic approximation level), $f(\boldsymbol{y})$ can be roughly approximated by $f(\boldsymbol{y}')$ and its gradient $\nabla f(\boldsymbol{y})^{\mathsf{T}}$ and $\nabla f(\boldsymbol{y}')^{\mathsf{T}}$, corresponding approximation quality can be measured by $\upsilon \|\boldsymbol{y}' - \boldsymbol{y}\|^3$. It is also observed that, if we want to approximate to $f(\boldsymbol{y})$ more accurately (e.g., at the level of $\upsilon \|\boldsymbol{y}' - \boldsymbol{y}\|^3$ in $\|\boldsymbol{y}' - \boldsymbol{y}\|$), higher order approximation item should be adopted (e.g., $\nabla f(\boldsymbol{y})^{\mathsf{T}}$ and $\nabla f(\boldsymbol{y}')^{\mathsf{T}}$), namely, the more information of $f(\boldsymbol{y})$ are explored, the more approximation we would achieve.

3.2 Two-Layers Coordinate Coding

Before defining two-layers coordinate coding, let's review the single-layer coordinate coding defined in [4] for its close relationship with Two-layers LCC.

Definition 2 (Single-layer Coordinate Coding). *A single-layer coordinate coding is a pair* $(\gamma^1, \boldsymbol{C}^1)$*, where* $\boldsymbol{C}^1 \subset \mathbb{R}^m$ *is a set of anchor points to* $\boldsymbol{y}$ *(aka basis functions in* $\boldsymbol{C}^1$*), and* $\boldsymbol{\gamma}^1$ *is a map of* $\boldsymbol{y} \in \mathbb{R}^m$ *to* $\gamma^1(\boldsymbol{y}) \in \mathbb{R}^{|\boldsymbol{C}^1|}$ *such that* $\left[\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right]_{\boldsymbol{v}\in\boldsymbol{C}^1} \in \mathbb{R}^1$ *and* $\sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) = 1$*. It induces the following physical approximation of* $\boldsymbol{y}$ *in* $\mathbb{R}^m$*:*

$$h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y}) \triangleq \boldsymbol{y}' = \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\boldsymbol{v}, \tag{6}$$

where, for conciseness, let's align all $\boldsymbol{\gamma}^1_{\boldsymbol{v}}(\boldsymbol{y})$ into a column vector: $\gamma^1(\boldsymbol{y}) = [\gamma^1_{\boldsymbol{v}_1}, \gamma^1_{\boldsymbol{v}_2}, \cdots, \gamma^1_{\boldsymbol{v}_{|\boldsymbol{C}^1|}}]^\mathsf{T} \in \mathbb{R}^{|\boldsymbol{C}^1|}$. In fact, the pre-image $\boldsymbol{y}$ is mapped into the image $\boldsymbol{y}'$ by the mapping $\boldsymbol{\gamma}^1$. Following the line of research, here we introduce two-layers coordinate coding form, accordingly:

Definition 3 (Two-layers Coordinate Coding). *A two-layers coordinate coding is two pairs with close relationship* $(\gamma^1, \boldsymbol{C}^1)$ *and* $(\boldsymbol{\gamma}^{2,\boldsymbol{v}}, \boldsymbol{C}^{2,\boldsymbol{v}})$*, where* $\boldsymbol{C}^{2,\boldsymbol{v}} \subset \mathbb{R}^m$ *is a set of anchor points to* $\boldsymbol{v}$ *rather than* $\boldsymbol{y}$*, and* $\boldsymbol{\gamma}^{2,\boldsymbol{v}}$ *is a map of* $\boldsymbol{v} \in \mathbb{R}^m$ *to* $\boldsymbol{\gamma}^{2,\boldsymbol{v}}(\boldsymbol{v}) \in \mathbb{R}^{|\boldsymbol{C}^{2,\boldsymbol{v}}|}$ *such that* $\left[\boldsymbol{\gamma}^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})\right]_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \in \mathbb{R}^1$ *and* $\sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v}) = 1$*. It induces the following physical approximation of* $\boldsymbol{v}$ *in* $\mathbb{R}^m$*:* $\boldsymbol{v}' = \sum\limits_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})\boldsymbol{u}$*, and corresponding two-layers approximation of* $\boldsymbol{y}$ *in* $\mathbb{R}^m$*:*

$$h_{\boldsymbol{\gamma}^{2,\boldsymbol{v}},\boldsymbol{C}^{2,\boldsymbol{v}}}(\boldsymbol{y}) \triangleq \boldsymbol{y}'' = \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})\boldsymbol{u}]. \tag{7}$$

For conciseness, let's rearrange all $\boldsymbol{\gamma}^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})$ into a column vector: $\boldsymbol{\gamma}^{2,\boldsymbol{v}}(\boldsymbol{v}) = [\gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}_1}(\boldsymbol{v}), \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}_2}(\boldsymbol{v}), \cdots, \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}_{|\boldsymbol{C}^{2,\boldsymbol{v}}|}}(\boldsymbol{v})]^\mathsf{T} \in \mathbb{R}^{|\boldsymbol{C}^{2,\boldsymbol{v}}|}$. The condition $\sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v}) = 1$ is shift-invariance requirement, which means the coding $\boldsymbol{v}'$ should remain the same if we use a different origin of the $\mathbb{R}^m$ coordinate system for representing $\boldsymbol{v}$.

Lemma 1 (Single-layer Linearization). *Let* f *be a* $(\alpha, \beta, \upsilon)$ *Lipschitz smooth function and* $(\gamma^1, \boldsymbol{C}^1)$ *an arbitrary single-layer coordinate coding on* $\mathbb{R}^m$*. For all* $\boldsymbol{y} \in \mathbb{R}^m$*:*

$$\begin{aligned}\left|f(\boldsymbol{y}) - \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) f(\boldsymbol{v})\right| &\le \alpha \left\|\boldsymbol{y} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\| + \beta \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \left[\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \left\|\boldsymbol{v} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\|^2\right] \\ &= \alpha \left\|\boldsymbol{y} - \boldsymbol{y}'\right\| + \beta \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \left[\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \left\|\boldsymbol{v} - \boldsymbol{y}'\right\|^2\right].\end{aligned} \tag{8}$$

A nonliear function $f(\boldsymbol{y})$ in $\mathbb{R}^m$ could be approximated by a linear function $\sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) f(\boldsymbol{v})$ with respect to $h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}$, i.e. the linear representation of $\boldsymbol{y}$, where $[f(\boldsymbol{v})]_{\boldsymbol{v}\in\boldsymbol{C}^1}$ is the set of coefficients viewed as unknown vectors and estimated from data using a standard learning method such as SVM.

The quality of this approximation is bounded by the right side of the inequation, which has two terms: the first term $\left\|\boldsymbol{y} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\|$ indicates the residual should be as small as possible; the second term suggests that $\sum_{\boldsymbol{v}\in\boldsymbol{C}^1} \gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\boldsymbol{v}$ should be localized, that is the sum of weighted distance between $\boldsymbol{y'}$ and each anchor point $\boldsymbol{v} \in \boldsymbol{C}^1$ should be as small as possible. The first term encourages the best approximation $\boldsymbol{y'}$ of $\boldsymbol{y}$, in particular, as illustrated in [4], for a smooth manifold, one can choose appropriate anchor points $\boldsymbol{C} \in \mathbb{R}^{\left|\boldsymbol{C}^1\right|}$ so that the first layer linearization could achieve local linear approximation power. While please note that, this approximation power is guaranteed under the precondition that we have to find enough descriptive anchor points to minimize the first term $\left\|\boldsymbol{y} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\|$, however, these anchor points are usually insufficient and noise-polluted in practical. The motivation of this paper is just to find this approximation at the second-layer level that provides an opportunity for zooming into each single basis $\boldsymbol{v} \in \mathbb{R}^{\left|\boldsymbol{C}^1\right|}$ of the first layer for finer local details, in order to finally incorporate more details about f extracting from the second layer and improve the approximation quality. Along the lines of researches, we principled generalize it to the two-layers structure, which is illustrated in the following lemma:

Lemma 2 (Two-layers Linearization). *Let f be a $(\alpha, \beta, \upsilon)$ Lipschitz smooth function and $(\boldsymbol{\gamma}^2, \boldsymbol{C}^2) = \left\{(\boldsymbol{\gamma}^1, \boldsymbol{C}^1)\right\} \cup \left\{(\boldsymbol{\gamma}^{2,v}, \boldsymbol{C}^{2,v}) : \boldsymbol{v} \in \boldsymbol{C}^1\right\}$ be an arbitrary two-layer coordinate coding on $\mathbb{R}^m$. For all $\boldsymbol{y} \in \mathbb{R}^m$:*

$$
\begin{aligned}
&|f(\boldsymbol{y}) - \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v}) f(\boldsymbol{u})]| \\
&\leq \alpha_1 \left\|\boldsymbol{y} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\| + \beta_1 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \left\|\boldsymbol{v} - h_{\boldsymbol{\gamma}^1,\boldsymbol{C}^1}(\boldsymbol{y})\right\|^2] \\
&\quad +\alpha_2 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| ||\boldsymbol{v} - \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})\boldsymbol{u}||] \\
&\quad +\beta_2 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})| \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \left|\gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}\right| ||\boldsymbol{u} - \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v})\boldsymbol{u}||^2] \\
&= \alpha_1 \left\|\boldsymbol{y} - \boldsymbol{y}'\right\| + \beta_1 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \left\|\boldsymbol{v} - \boldsymbol{y}'\right\|^2] \\
&\quad +\alpha_2 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \left\|\boldsymbol{v} - \boldsymbol{v}'\right\|] \quad + \beta_2 \sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\left|\gamma^1_{\boldsymbol{v}}(\boldsymbol{y})\right| \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \left|\gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}\right| \left\|\boldsymbol{u} - \boldsymbol{v}'\right\|^2].
\end{aligned}
\tag{9}
$$

On the left side of the inequation, a nonlinear function $f(\boldsymbol{y})$ in $\mathbb{R}^m$ is approximated by a linear function: $\sum_{\boldsymbol{v}\in\boldsymbol{C}^1} [\gamma^1_{\boldsymbol{v}}(\boldsymbol{y}) \sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}} \gamma^{2,\boldsymbol{v}}_{\boldsymbol{u}}(\boldsymbol{v}) f(\boldsymbol{u})]$ with respect to $h_{\boldsymbol{\gamma}^{2,\boldsymbol{v}},\boldsymbol{C}^{2,\boldsymbol{v}}}$, where $[f(\boldsymbol{u})]_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}}$ is the set of coefficients, which could also be estimated using the same approach as in the single-layer linearization. The quality of this approximation is bounded by the right side of the equation: the first two terms have the same meaning as the ones introduced in the Lemma 1. The third term indicates the weighted residual should be as small as possible, i.e., $\boldsymbol{v'} \in \mathbb{R}^{\left|\boldsymbol{C}^{2,\boldsymbol{v}}\right|}$ should be close to its preimage $\boldsymbol{v} \in \boldsymbol{C}^1$; while the forth term encourages localization in the coding $\boldsymbol{v'}$ of $\boldsymbol{v}$.

In addition, we also make a critical observation that a nonlinear function $f(\boldsymbol{y})$ in $\mathbb{R}^m$ could be approximated by a linear function with two-layers structure: at first layer, the original $f(\boldsymbol{y})$ is divided into $\left|\boldsymbol{C}^1\right|$ components: $f(\boldsymbol{v}_1),\cdots,f(\boldsymbol{v}_{|\boldsymbol{C}^1|})$, which are then linearly combined to compose $f(\boldsymbol{y}')$, and if some preconditions are guaranteed (i.e., the first two terms of the bound in Lemma 2), the $1st$ order approximation $f(\boldsymbol{y}')$ could achieve a satisfactory result; while in second layer, each $f(\boldsymbol{v})$ is further divided into $\left|\boldsymbol{C}^{2,\boldsymbol{v}}\right|$ sub-components $f(\boldsymbol{u}_1),\cdots,f(\boldsymbol{u}_{|\boldsymbol{C}^{2,\boldsymbol{v}}|})$, which are then linearly combined to compose corresponding $f(\boldsymbol{v}')$. Then, all these various $f(\boldsymbol{v}_1'),\cdots,f(\boldsymbol{v}'_{|\boldsymbol{C}^1|})$ are delivered up to the first layer for finally composing $f(\boldsymbol{y}'')$ linearly, and if some preconditions are guaranteed (i.e., the remainders of the bound in Lemma 2), the $2nd$ order approximation $f(\boldsymbol{y}'')$ could achieve a more satisfactory result than its single-layer counterpart.

Moreover, the two-layer structure also incarnates the quality of computation saving, namely, each set of sub-bases $\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}$ corresponding to $\boldsymbol{v}$ with nonzero coefficients at the first layer could be calculated simultaneously, for instance, instead of fitting a single model with many atoms in the dictionary $\boldsymbol{C}\in\mathbb{R}^m$, two-layer hierarchical structure need only fit a dozens of small local system with grouped atoms in parallel, which dramatically improves the computational complexity. So in the next section, we will pay more attention to practical computational procedure.

4 Two-Layers Coding Formulation

This section will discuss practical computational procedure. In the spirit of reducing the error and encouraging the locality at different levels, a hierarchical method accommodating the underlying intuition is designed.

4.1 First-Layer Formulation

Let $\boldsymbol{Y}$ be a set of m-dimensional local descriptors extracted from a sampled data, i.e., $\boldsymbol{Y}=[\boldsymbol{y}_1,\cdots,\boldsymbol{y}_N]\in\mathbb{R}^{m\times N}$. Given a first-layer codebook with D_1 entries, $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_{D_1}]\in\mathbb{R}^{m\times D_1}$, first-layer coding schemes will convert each local descriptor $\boldsymbol{y}_i$ into a D_1-dimensional code $\boldsymbol{\gamma}_i^1=\left[\gamma_i^1(\boldsymbol{v}_1),\gamma_i^1(\boldsymbol{v}_2),\cdots,\gamma_i^1(\boldsymbol{v}_{D_1})\right]^{\mathsf{T}}\in\mathbb{R}^{D_1}$, hence, arrange all codes into a matrix: $\boldsymbol{\gamma}^1=\left[\boldsymbol{\gamma}_1^1,\boldsymbol{\gamma}_2^1,\cdots,\boldsymbol{\gamma}_N^1\right]\in\mathbb{R}^{D_1\times N}$. Specifically, each code could be obtained using the following optimization form:

$$\min_{\boldsymbol{\gamma}_i^1}\left[\frac{1}{2}\left\|\boldsymbol{y}_i-\boldsymbol{V}\boldsymbol{\gamma}_i^1\right\|_2^2+\beta\left\|\boldsymbol{\gamma}_i^1\odot\boldsymbol{d}_i^1\right\|_1\right]\quad s.t.\quad \mathbf{1}^{\mathsf{T}}\boldsymbol{\gamma}_i^1=1 \tag{10}$$

where $\boldsymbol{d}_i^1\in\mathbb{R}^{D_1}$ is a distance vector, each item of which measures the distance between $\boldsymbol{y}_i$ and $\boldsymbol{v}_i$, and $\odot$ denotes the element-wise multiplication, which enables corresponding items of both vectors ($\boldsymbol{\gamma}_i^1$ and $\boldsymbol{d}_i^1$) to multiply. Typically, $\boldsymbol{d}_i^1$ can be obtained using ℓ_2 norm, that is $\boldsymbol{d}_i^1=[\|\boldsymbol{y}_i-\boldsymbol{v}_1\|_2,\|\boldsymbol{y}_i-\boldsymbol{v}_2\|_2,\cdots,\|\boldsymbol{y}_i-\boldsymbol{v}_{D_1}\|_2]^{\mathsf{T}}$. The constraint $\mathbf{1}^{\mathsf{T}}\boldsymbol{\gamma}_i^1=1$ follows the shift-invariant requirements of the two-layer code.

4.2 Second-Layer Formulation

At the second layer, we would further refine each basis $\boldsymbol{v}$ belonging to the first layer. Concretely, the third and fourth terms of the bound in Lemma 2 specify how we refine each basis vector $\boldsymbol{v}$ at the second layer, during which, more information about the gradient of f: $\nabla f(\boldsymbol{v}')^{\mathsf{T}}$ are then be incorporated. Before optimizing the third and fourth terms, both of them are further transformed into the following form:

$$\begin{aligned}&\min[\sum_{\boldsymbol{v}\in\boldsymbol{C}^1}[\left|\gamma_{\boldsymbol{v}}^1(\boldsymbol{y})\right|(\|\boldsymbol{v}-\boldsymbol{v}'\|+\sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}}\left|\gamma_{\boldsymbol{u}}^{2,\boldsymbol{v}}\right|\|\boldsymbol{u}-\boldsymbol{v}'\|^2)]]\\&\leq\sum_{\boldsymbol{v}\in\boldsymbol{C}^1}\min[\left|\gamma_{\boldsymbol{v}}^1(\boldsymbol{y})\right|(\|\boldsymbol{v}-\boldsymbol{v}'\|+\sum_{\boldsymbol{u}\in\boldsymbol{C}^{2,\boldsymbol{v}}}\left|\gamma_{\boldsymbol{u}}^{2,\boldsymbol{v}}\right|\|\boldsymbol{u}-\boldsymbol{v}'\|^2)],\end{aligned}\tag{11}$$

which indicates the problem could be further divided into a set of small models at the second layer, and thus be tackled individually. In addition, fitting the small models can be done in parallel, from which two-layer coding is benefited.

Therefore, this leads to the following formulation for each small model:

$$\min\left[\tfrac{1}{2}\left\|\boldsymbol{v}_i-\boldsymbol{U}\boldsymbol{\gamma}_i^{2,\boldsymbol{v}}\right\|_2^2+\beta\left\|\boldsymbol{\gamma}_i^{2,\boldsymbol{v}}\odot\boldsymbol{d}_i^2\right\|_1\right]\quad s.t.\quad \mathbf{1}^{\mathsf{T}}\boldsymbol{\gamma}_i^{2,\boldsymbol{v}}=1,\tag{12}$$

where $\boldsymbol{d}_i^2\in\mathbb{R}^{D_2}$ is also a distance vector recording the distance between $\boldsymbol{v}_i$ and each atom in the dictionary matrix $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{D_2}]$; $\boldsymbol{v}_i\in\boldsymbol{V}$ is one of basis vectors adopted in the representation of $\boldsymbol{y}_i$ at the first layer, which could then be augmented into a D_2-dimensional code $\boldsymbol{\gamma}_i^{2,\boldsymbol{v}}=\left[\gamma_i^{2,\boldsymbol{v}}(\boldsymbol{u}_1),\gamma_i^{2,\boldsymbol{v}}(\boldsymbol{u}_2),\cdots,\gamma_i^{2,\boldsymbol{v}}(\boldsymbol{u}_{D_2})\right]^{\mathsf{T}}\in\mathbb{R}^{D_2}$, hence after rearranging each code corresponding to each basis vector $\boldsymbol{v}$ into a matrix, we obtain the coding matrix for all $\boldsymbol{v}$ adopted in the representation of $\boldsymbol{y}_i$: $\boldsymbol{\gamma}^{2,\boldsymbol{v}}=\left[\boldsymbol{\gamma}_1^{2,\boldsymbol{v}},\boldsymbol{\gamma}_2^{2,\boldsymbol{v}},\cdots,\boldsymbol{\gamma}_{|\boldsymbol{C}_{\boldsymbol{y}_i}^1|}^{2,\boldsymbol{v}}\right]\in\mathbb{R}^{D_2\times|\boldsymbol{C}_{\boldsymbol{y}_i}^1|}$. This matrix comprises the second layer coding for each local descriptor $\boldsymbol{y}_i$ indirectly. Furthermore, the final form of two-layers coding for each $\boldsymbol{y}_i$ could be obtained by integrating each single-layer coding organically. Specifically, each item (e.g., the jth item) in the first layer coding $\boldsymbol{\gamma}_i^1$ is augmented into a vector, $\boldsymbol{\gamma}_i=\left[\gamma_i^1(\boldsymbol{v}_j),\ \gamma_i^1(\boldsymbol{v}_j)[\gamma_j^{2,\boldsymbol{v}}(\boldsymbol{u}_1),\gamma_j^{2,\boldsymbol{v}}(\boldsymbol{u}_2),\cdots,\gamma_j^{2,\boldsymbol{v}}(\boldsymbol{u}_{D_2})]\right]^{\mathsf{T}}\in\mathbb{R}^{1+D_2}$, which constitutes the final form of two-layer coding in $\mathbb{R}^{D_1\times(1+D_2)}$.

5 Experiment Verification

We present experiments on: the Extended YaleB [8], Caltech101 [9], the MNIST [4], and a realistic robot system to evaluate the every aspect of the proposed strategy.

5.1 Quantitative Results

Due to the space limitation, we omit brief introduction of these popular databases, for more details, one can refer to the references marked behind them.

Table 1. Description of experimental settings.

Database	Training samples	$\boldsymbol{V}$	$\boldsymbol{U}$	Train\Test
eYaleB	32	570	64	2000\ 600
Caltech101	5,10,20,30	510,1020,2040,3060	64	8230\ 914
MNIST	3000	500	64	60000\ 10000

Table 2. Recognition results and computation time comparisons on the extended YaleB database.

Method	LLC [7]	SRC [10]	LC-KSVD1 [11]	LC-KSVD2 [11]	ITDL [12]	Ours (15 per person)
Included	0.9070	0.8050	0.9450	0.9500	0.9539	**0.9813**
Excluded[1]	0.9670	0.8670	0.9830	0.9880	0.9886	**0.9903**
Average Time (ms)	-	11.22	0.52	0.49	-	**55.90**

1: This column is the result when 10 poor-quality images excluded for each class.

Table 3. Recognition results on the MNIST database.

Method	Laplacian Eigenmap [13]	Deep Belief Network [14]	LLE [15]	LCC [4]	DCN [6]	RGF [16]	Ours
Accuracy(%)	0.9727	0.9810	0.9762	0.9810	0.9815	0.9809	**0.9857**

Table 4. Recognition results on Caltech101 database.

Training samples	LLC [7]	SRC [10]	K-SVD [2]	LC-KSVD2 [11]	SSC [17]	Ours
5	0.5115	0.4880	0.4980	0.5400	0.5560	**0.5783**
10	0.5977	0.6010	0.5980	0.6310	0.6550	**0.6572**
20	0.6774	0.6770	0.6870	0.7050	0.7620	**0.7535**
30	0.7344	0.7070	0.7320	0.7360	0.7760	**0.7989**

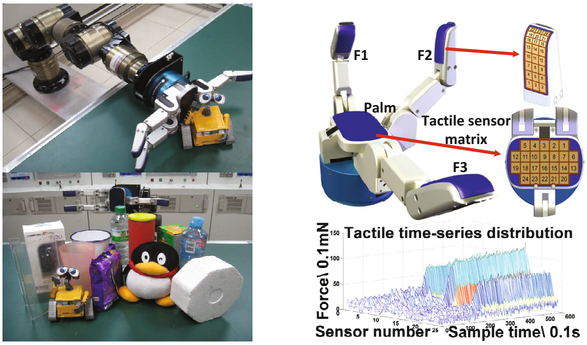
Table 5. Recognition results (computation time (ms) for classifying a test image) on the Caltech101 dataset (varying dictionary size).

Dictionary size	510	1020	2040	3060
SRC [10]	0.48 (173.44)	0.60 (343.12)	0.67 (662.40)	0.71 (987.55)
LC-KSVD1 [11]	0.71 (0.59)	0.72 (1.09)	0.72 (2.21)	0.74 (3.50)
LC-KSVD2 [11]	0.72 (0.54)	0.73 (0.98)	0.73 (1.94)	0.74 (3.17)
Ours	**0.72** (99.85)	**0.75** (196.52)	**0.79** (384.23)	**0.80** (595.93)

For fair comparison, we adopt the same experiment setups suggested by the homepages of the databases and related literatures [7,10–12,17], etc. We summarize the key details of experimental settings in Table 1. From Table 2 to Table 5, it is consistently observed that our method exhibits a prominent recognition accuracy in all databases, even with fewer training samples and smaller dictionary size. The main reason lies in, two-layers strategy fully exploits the intrinsic structure of the manifold where datapoints reside in, and incorporates more information about the nonlinear function f in anchor points of each layers, which could greatly benefit their approximation power, especially in the situation of fewer training samples.

5.2 Applications in Robotics

Since the proposed coding strategy exhibits outstanding performance on popular databases, how it works in practical use, especially in noise polluted conditions? This section reveals applications of our theoretical results on a real robot system. For better understanding, we briefly present the experimental settings and tasks as follows: we have employed Barrett^{TM}robot hand fixed on a 7-DOF Schunk^{TM}modular robot to perform a task of multi-objects grasping and classification, see Figure 2(a). One of features of this robotic system lies in its large amount of informative tactile data provided by the tactile sensor matrix mounted on the fingers and palm, see Figure 2(b), illustrating distributions of tactile sensors mounted on each fingertip (F1, F2 and F3) and the palm of Barrett^{TM}hand, and each part samples the force variation in the contact area. Distributions and magnitude of tactile time-series have the ability to reflect meticulous state of fingertips and objects, therefore we could infer target class from it. While this type of data has a major disadvantage: it is vulnerable to noise, which greatly challenges signal processing, see the bottom of Figure 2(b).



(a) Targets grasping. (b) Sensors and data.

Fig. 2. Targets grasping and distributions of tactile data.

How to fully exploit this informative data for an accurate inference is a challenging problem, which provides just an nice opportunity for our proposed representation approach. To verify the proposed strategy, we repeat grasping dozens of targets with various shape and material, and record the tactile time-series; then various coding strategies are further employed to describe these signals and finally classified by linear SVM. Classification accuracy comparison is presented in Table 6. It is observed that our two-layers coding strategy outperforms sparse coding and LLC methods at every signal-to-noise (SNR) level, except at 5dB SNR, showing that it is capable of resisting the noise even in some extreme case of noise level 10dB or 15dB SNR.

Table 6. Recognition results on robotic testbed with varying noise levels.

Noise level	5dB	10dB	15dB	20dB	30dB	40dB
LLC[7]	0.70	0.74	0.78	0.81	0.83	0.85
SRC[10]	**0.71**	0.74	0.83	0.84	0.88	0.89
Ours	**0.71**	**0.76**	**0.85**	**0.87**	**0.89**	**0.91**

6 Conclusions

This paper systematically proposes a principled extension of the traditional single-layer sparse coding scheme for high dimensional nonlinear learning. The proposed method is viewed as generalized local linear function approximation, but can achieve higher approximation power due to additional gradient information about the nonlinear function included. The main advantages of two-layers coding is that it can potentially achieve better performance due to the introduction of the second layer, which incorporates abundant information about nonlinear function. Experiment evaluations on both popular benchmarks and robotic application further confirm our analysis.

References

1. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (2006)
2. Aharon, M., Elad, M., Bruckstein, A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing **54**(11), 4311–4322 (2006)
3. Kouskouridas, R., Charalampous, K., Gasteratos, A.: Sparse pose manifolds. Autonomous Robots **37**(2), 191–207 (2014)
4. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems, pp. 2223–2231 (2009)
5. Yu, K., Zhang, T.: Improved local coordinate coding using local tangents. In: Proceedings of the 27th International Conference on Machine Learning, pp. 1215–1222 (2010)
6. Lin, Y., Tong, Z., Zhu, S., Yu, K.: Deep coding network. In: Advances in Neural Information Processing Systems, pp. 1405–1413 (2010)
7. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3360–3367 (2010)
8. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(6), 643–660 (2001)
9. Fei-Fei, L., Fergus, R., Perona, P.: 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 (2007)
10. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 210–227 (2009)

11. Jiang, Z., Lin, Z., Davis, L.S.: Learning a discriminative dictionary for sparse coding via label consistent k-svd. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1697–1704 (2011)
12. Qiu, Q., Patel, V.M., Chellappa, R.: Information-theoretic dictionary learning for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(11), 2173–2184 (2014)
13. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
14. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
15. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
16. Fan, M., Zhang, X., Lin, Z., Zhang, Z., Bao, H.: A regularized approach for geodesic-based semisupervised multimanifold learning. IEEE Transactions on Image Processing **23**(5), 2133–2147 (2014)
17. Oliveira, G.L., Nascimento, E.R., Vieira, A.W.: Montenegro Campos, M.F.: Sparse spatial coding: A novel approach to visual recognition. IEEE Transactions on Image Processing **23**(6), 2719–2731 (2014)