Neurocomputing 157 (2015) 61–69

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Online dictionary learning for Local Coordinate Coding with Locality Coding Adaptors

Junbiao Pang [a], Chunjie Zhang [b], Lei Qin [c], Weigang Zhang [d], Laiyun Qing [b], Qingming Huang [a,b,c,*], Baocai Yin [a,**]

[a] Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation, Beijing University of Technology, No. 100 Pingleyuan Road, Chaoyang District 100124, China
[b] School of Computer and Control Engineering, University of Chinese Academy of Sciences, No. 19 Yuquan Road, Shijingshan District, Beijing 100049, China
[c] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, No. 6 Kexueyuan South Road, Haidian District, Beijing 100190, China
[d] School of Computer Science and Technology, Harbin Institute of Technology at Weihai, No. 2 West Wenhua Road, Weihai 26209, China

ARTICLE INFO

Article history:
Received 20 January 2014
Received in revised form
31 October 2014
Accepted 17 January 2015
Communicated by Steven Hoi
Available online 2 February 2015

Keywords:
Local Coordinate Coding
Surrogate function
Locality Coding Adaptor
Large scale problem
Online training

ABSTRACT

Dictionary in Local Coordinate Coding (LCC) is important to approximate a non-linear function with linear ones. Optimizing dictionary from predefined coding schemes is a challenge task. This paper focuses on learning dictionary from two Locality Coding Adaptors (LCAs), i.e., locality Gaussian Adaptor (GA) and locality Euclidean Adaptor (EA), for large-scale and high-dimension datasets. Online dictionary learning is formulated as two cycling steps, local coding and dictionary updating. Both stages scale up gracefully to large-scale datasets with millions of data. The experiments on different applications demonstrate that our method leads to a faster dictionary learning than the classical ones or the state-of-the-art methods.

1. Introduction

Local Coordinate Coding (LCC) [1] is a general framework that uses linear functions to approximate any non-linear Lipschitz smooth one. LCC generally consists of two key components: (1) the coding schemes that define the local coordinates [2]; and (2) a dictionary (data points) which consists of the local coordinates. LCC has been successfully applied to many challenging problems, e.g., approximating non-linear kernels [3], feature learning in multi-class classification [4].

The problem to learn dictionary for LCC, especially for high-dimension visual data, is that time complexity grows quadratically with both the dictionary size and the dimension of data. Because sparse coding [5,6] is usually used to lean dictionary [1]. For a large-scale dataset with millions of samples, the time cost of this sparse coding-based approach [1] becomes unacceptable. For instance, on a single-core 2.6 GHz machine, sparse coding takes about a week to lean 1000 items of a dictionary from about one million samples via feature-sign search [7].

To avoid the sparse coding during learning dictionary for LCC, Locality Coding Adaptors (LCAs) [4,2] are proposed to replace the locality error in LCC. The dictionary size in real applications seemingly increases explosively for high-dimension data, as items in a dictionary should be "local enough" to encode a sample. For LCAs, however, one of the recent results [2] discovers that both locality Gaussian Adaptor (GA) [4] and locality Euclidean Adaptor (EA) [2,8] have no relation with the dimension of data. Therefore, the motivation behind this paper is to fast and accurately learn dictionary for LCC with LCAs.

The key notation of our solution is that dictionary can be fast computed with both surrogate function [9] and warm restart technique [10]. Rather than adopting Stochastic Gradient Descent (SGD) (which requires to tune a learning speed), we instead use the surrogate method [9,11] which aggregates the past information computed during the previous steps with warm restart. The advantage

* Corresponding author at: Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation, Beijing University of Technology, No. 100 Pingleyuan Road, Chaoyang District 100124, China.
** Corresponding author.
E-mail addresses: junbiao_pang@bjut.edu.cn (J. Pang), cjzhang@jdl.ac.cn (C. Zhang), lqin@jdl.ac.cn (L. Qin), wgzhang@jdl.ac.cn (W. Zhang), lyqing@ucas.ac.cn (L. Qing), qmhuang@jdl.ac.cn (Q. Huang), ybc@bjut.edu.cn (B. Yin).

http://dx.doi.org/10.1016/j.neucom.2015.01.035
0925-2312/

of warm restart is that a good initialization is supplied when dictionary is learned with the block-wise coordinate descent [12]. This learning scheme is not only significantly faster than the batch alternatives, but also hopes to avoid tuning hyper-parameters in SGD, e.g., learning speed.

The core contributions of this paper can be summarized as follows: technically, we introduce an online dictionary learning method for LCC with LCAs. Our learning approach achieves approximate 100 times faster than the batch one [8] on large-scale datasets. Besides, the theoretical justification on the convergence of the proposed algorithm is presented.

In the next section, related work is briefly summarized. Section 3 introduces problem of learning dictionary for LLC with LCAs. Section 4 first outlines the dictionary learning algorithm and details two cycling steps. After that are the experiment and conclusion sections.

2. Related work

The seemingly most similar work to LCC may be dictionary learning in sparse coding [13,14]: adding different constraints into the reconstruction losses. However, the goal of sparse coding is to represent a signal approximately as the globally linear combination of a small number of the overcomplete dictionary. Given data $\mathbf{x}_i$ ($\mathbf{x}_i \in \mathbb{R}^D$) and dictionary $\mathbf{V}=[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M]$ ($\mathbf{V} \in \mathbb{R}^{D\times M}$), sparse coding seeks a linear reconstruction of the given data $\mathbf{x}_i$ as $\mathbf{x}_i = \gamma_{i1}\mathbf{v}_1 + \gamma_{i2}\mathbf{v}_2 + \cdots + \gamma_{iM}\mathbf{v}_M$. The reconstruction coefficients $\gamma_i = [\gamma_{i1}, \ldots, \gamma_{iM}]^T$ ($\gamma_i \in \mathbb{R}^M$) are sparsity, requiring only a small fraction of entries in γ_i are nonzeros. Denoting $\|\gamma\|_0$ as the number of nonzero entries of the vector γ, sparse coding can be formulated as follows:

$$\begin{aligned} \min_{\gamma_i} \quad & \|\gamma_i\|_0 \\ \text{s.t.}: \quad & \mathbf{x}_i = \mathbf{V}\gamma_i. \end{aligned} \tag{1}$$

However, the minimization of ℓ_0 norm is an NP-hard problem. Recent research usually formulates the sparse coding problem as the minimization of ℓ_1 norm of the reconstruction coefficients. The objective of sparse coding can be reformulated as follows [7,15]:

$$\min_{\gamma_i, \mathbf{V}} \|\mathbf{x}_i - \mathbf{V}\gamma_i\|^2 + \lambda\|\gamma_i\|_1, \tag{2}$$

The first term in (2) is the reconstruction loss, and the second term is used to control the sparsity. λ is the tradeoff parameter used to balance the sparsity and the reconstruction error.

While the locality of LCC tends to bring sparsity into local coding, as only the items in a dictionary closing to the test input would be given more weights. The objective of LCC is formulated as follows [1]:

$$\left| f(\mathbf{x}_i) - \sum_m \gamma_{im} f(\mathbf{v}_m) \right| \le \alpha \left\|\mathbf{x}_i - \mathbf{V}\gamma_i\right\|^2 + \beta \sum_m |\gamma_{im}| \left\|\mathbf{v}_m - \mathbf{V}\gamma_i\right\|^{1+p}, \tag{3}$$

where a nonlinear function $f(\mathbf{x}_i)$ is approximated by a set of linear ones $f(\mathbf{v}_m)$, $\mathbf{V}=[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M]$ ($\mathbf{V} \in \mathbb{R}^{D\times M}$) is the dictionary, γ_{im} are the local coding of data $\mathbf{x}_i$ based on the point $\mathbf{v}_m$, and α and β are the tradeoff factors to balance between the reconstruction error $\left\|\mathbf{x}_i - \mathbf{V}\gamma_i\right\|^2$ and the locality one $|\gamma_i| \left\|\mathbf{v}_m - \mathbf{V}\gamma_i\right\|^{1+p}$. Eq. (3) indicates that LCC locally encodes each sample to obtain the non-linear approximation ability. In contrast, the dictionary in sparse coding (2) does not favor this choice. Therefore, the motivation between sparse coding and LCC is totally different.

Learning dictionary for LCC in (3) has to face the non-smooth optimization $|\gamma_i| \left\|\mathbf{v}_m - \mathbf{V}\gamma_i\right\|^{1+p}$ and the choice of hyper-parameter p. To avoid these difficulties, LCAs are proposed to replace the locality error as follows:

$$\left| f(\mathbf{x}_i) - \sum_m \gamma_{im} f(\mathbf{v}_m) \right| \le \alpha \left\|\mathbf{x}_i - \mathbf{V}\gamma_i\right\|^2 + \beta \|\mathbf{p}_i \odot \gamma_i\|^2, \tag{4}$$

where the operation $\odot$ represents the element-wise multiplication, and $\mathbf{p}_i$ ($\mathbf{p}_i \in \mathbb{R}^M$) are LCAs. The second term in (4) enforces local coding γ_{im} to have a similar locality of LCAs [2]. Therefore, instead of learning dictionary for LCC, learning dictionary for LCC with LCAs has several advantages: (1) the smooth objective function in LCC with LCAs avoids the non-smooth optimization in LCC; (2) the dictionary size of LCC with LCAs has no relation with the dimension of data [2]. Concretely, LCC with LCAs avoids the sparse coding problem in LCC with the complexity $\mathcal{O}(DMs + Ds^2)$, where s is the number of the nonzero coefficients, if a Cholesky-based implementation of LASSO/LARS problem [5] is adopted. Moreover, LCC with LCAs turns LCC (3) into convex objective functions when one of the parameters $\{\gamma_i, \mathbf{V}\}$ is fixed (see Section 4.4 for the detailed analysis).

Dictionary learning in LCC with LCAs is in most cases considered as vector quantization (VQ). However, a large part of the classical approaches in VQ barely handle a predefined locality. For example, [3] uses k-means to participate the data space with Euclidean distance (which can be considered as a special case of EA [2]). Other methods define a special locality according to the adopted VQ, e.g., [16] uses LASSO to solve a coding scheme with inverse Euclidean distance. These, however, lose flexibility to optimize dictionary for different LCAs.

Dictionary learning in LCC with LCAs alternates between two steps: local coding and dictionary updating. The local coding is sequentially learned for every sample, only requiring a limited computational cost. Dictionary updating by a batch training algorithm [8] has to process all samples in each iteration. Recently, [4] relaxes the objective function by ignoring LCAs, and learns dictionary by minimizing the reconstruction loss. However, the relaxed the objective function makes the learned dictionary obtain a suboptimal performance.

Inspired by the success of warm restart [10] in online sparse coding [11], our proposed method also applies this technique to update dictionary. Dictionary updating in online sparse coding minimizes the convex reconstruction loss in (2), while our approach optimizes both the reconstruction loss and the locality loss (4). On the other side, recently there has been a trend of introducing surrogate function into different tasks, and thus the optimization problem is viewed as finding a more approximate yet simple objective function [17,18,11]. To the best of our knowledge, this paper is first to apply the surrogated-based method to dictionary learning in LCC with LCAs.

3. Problem formulation

3.1. Dictionary learning with Locality Coding Adaptors

LCC is formulated as a constrained reconstruction problem, as the quality of the non-linear approximation ability is bounded by both the reconstruction and the locality (3). For a matrix $\mathbf{X}=[\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ with N data, the dictionary matrix $\mathbf{V}=[\mathbf{v}_1, \ldots, \mathbf{v}_M]$, LCC with LCAs can be formulated as the following problem [4]:

$$L(\mathbf{X}, \mathbf{V}) = \min_{\gamma_i, \mathbf{V}} \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{V}\gamma_i\|^2 + \lambda \|\mathbf{p}_i \odot \gamma_i\|^2 \tag{5}$$

$$\text{s.t.} \quad \gamma_i^T \mathbf{1} = 1, i = 1, \ldots, N, \tag{6}$$

where the vector $\mathbf{1}$ denotes the identity vector $[1, \ldots, 1]^T$, and the operation $\odot$ represents the element-wise multiplication, and $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iM}]^T$ ($\mathbf{p}_i \in \mathbb{R}^M$) are LCAs. p_{im} can be either GA [4] or EA [8,2]:

1. *Gaussian adaptor* (*GA*) presumes the relation among samples and dictionary as

$$p_{im} = \exp\left(\frac{\|\mathbf{v}_m - \mathbf{x}_i\|^2}{\sigma^2}\right) \tag{7}$$