Neurocomputing 184 (2016) 36–42

Contents lists available at ScienceDirect

Neurocomputing

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

A robust local sparse coding method for image classification with Histogram Intersection Kernel

Pan Li, Yang Liu *, Guojun Liu, Maozu Guo, Zhiyong Pan

School of computer science, Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Article history:
Received 22 January 2015
Received in revised form
13 May 2015
Accepted 14 July 2015
Available online 14 December 2015

Keywords:
Local sparse coding
Histogram Intersection Kernel
Image classification
Non-negative sparse coding

ABSTRACT

Local sparse coding methods have been shown to lead to increased performance in image classification when it takes histograms as inputs. These methods often use Euclidean (l_2) distance to learn the dictionary and encode the histograms. However, it has been shown that Histogram Intersection Kernel (HIK) is more effective to compare histograms. In this paper, we combine Histogram Intersection Kernel with local sparse coding. We implement dictionary learning and feature encoding on the mapping space that corresponds to the kernel. To encode the features, we propose two methods: one accurate method generating codes consisting of positive and negative values and one approximate method generating only non-negative values. Both of the two encoding methods run very fast. To verify our method, we conduct some experiments on two popular datasets: Caltech-101 and Caltech-256. The results show that the features extracted by our method are more discriminative than other methods and it reaches state-of-the-art result on Caltech-101 when taking single descriptor HOG as input. In addition, it shows that the codes with non-negative constraint are more effective than that without the constraint.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Sparse coding (SC) was first introduced in [1] for modeling the low-level sensory processing in human visual system. After that, many sparse coding methods have been proposed and applied in lots of fields, such as face recognition [2,3], image super-resolution [4], and data segmentation [5]. However, SC sometimes works unstably, which means that similar inputs will get dissimilar sparse codes, while dissimilar inputs may unexpectedly get similar sparse codes. Fig. 1 shows the case. Yu et al. [6] demonstrate that the sparse codes tend to be local and similar inputs should get similar codes. Based on this point, they propose a local sparse coding method (LSC) to implement local constraints. In this direction, Wang et al. [7] present another effective encoding scheme called Locality-constrained Linear Coding (LLC), which also implements locality by projecting each descriptor into its local-coordinate system. Gao et al. [8] propose another local sparse coding method, named Laplacian sparse coding; it introduces the Laplacian matrix into the loss function, and makes sure that the generated codes are local. Zheng et al. [9] propose a graph-regularized method to reserve locality, which is similar to [8].

Dictionary learning plays a very important role for many pipelines in computer vision. In the bag-of-visual-words framework [10], a dictionary is the whole vocabularies, and any patch of an image can be assigned to one vocabulary. In the SC framework, a learned dictionary shows the edges information for images [11]. Coates et al. [12] have shown that K-Means [13] is usually sufficient to learn the dictionary. In LLC, it shows two approaches to compute the dictionary, and one of them is K-Means, which computes the distances between inputs and cluster centers using Euclidean (l_2) distance in the E step, and updates cluster centers in M step. If we change l_2 distance by l_1 distance in E step, it will be a K-Medoids method [14]. Wu et al. [15] first use the Histogram Intersecting Kernel (HIK) [16] as the similarity metric for image descriptors to learn a dictionary, and the dictionary is very effective. HIK can be applied in dictionary learning, and it is interesting to exploit further whether HIK can be used in encoding step.

In this paper, we combine HIK with local sparse coding framework (LLC), which learns the dictionary and computes local sparse codes for input histograms on the mapping space that corresponds to HIK. Fig. 2 shows the main idea of our work. It first translates the inputs onto the mapping space, and then learns the dictionary, and finally encodes the inputs by its local coordinate system on the mapping space. As far as we can see, this approach is similar to kernel fisher discriminant (KFD) [17], or kernel principal component analysis (KPCA) [18], all of which compute the codes on the mapping space. The difference is that the final codes generated in our approach is over-complete, and sparse, while KFD or KPCA can only get compact, dense codes.

The remainder of the paper is organized as follows: in Section 2, we review the related works, including LLC and HIK. In Section 3, we discuss our framework, a robust local sparse coding method with HIK. In Section 4, it shows the experimental results. Finally, discussion and conclusion are made in Section 5.

* Corresponding author. Tel.: +86 13029840076.
E-mail addresses: yliu76@hit.edu.cn (Y. Liu), maozuguo@hit.edu.cn (M. Guo).

http://dx.doi.org/10.1016/j.neucom.2015.07.136
0925-2312/© 2015 Elsevier B.V. All rights reserved.

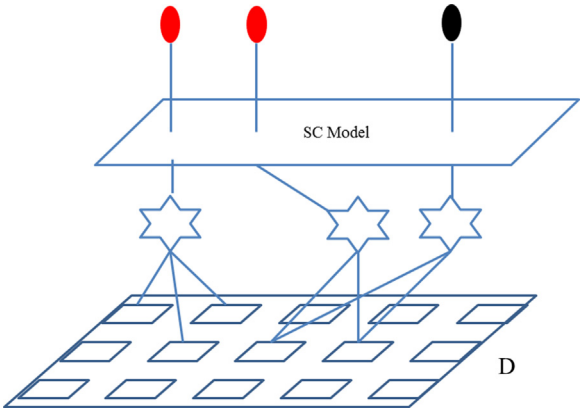

Fig. 1. Illustration of the instability of sparse coding. The plate D at the bottom is the dictionary and every polygon stands for one base. It shows that similar inputs (with the same color 'red') have very different active bases, while dissimilar inputs (with different colors 'red' and 'black') have similar active bases through SC model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

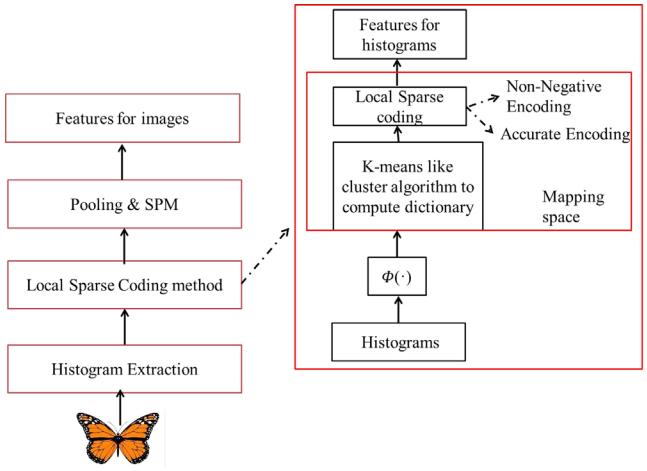

Fig. 2. Main idea of our work. The left side pipeline is one typical representation learning pipeline. Given a low-level image, it extracts the image histograms (e.g., SIFT [19], HOG [20]), and then encodes the histograms by local sparse coding method, followed by pooling and spatial pyramid matching (SPM) [21] to get the representation of one image. The right side box shows our novel idea of encoding image histograms. Dictionary learning and histogram encoding are done on the mapping space. At the encoding step, we propose two approaches, one with non-negative constraint and the other without non-negative constraint.

2. Related work

Our work combines LLC [7] and HIK [16]. LLC will be briefly described first, followed by details of HIK.

2.1. Locality-constrained Linear Coding

Locality-constrained Linear Coding method (LLC) is proposed by Wang et al. [7]. This method guarantees the locality of the input data.

Suppose that we have the local histograms for every image in hand (e.g. HOG [20]). LLC first uses these histograms to compute the dictionary $\boldsymbol{B}=[\boldsymbol{b}_1,..,\boldsymbol{b}_q]\in R^{d\times q}(d\ll q)$ by K-Means[1], where d is the dimension of every histogram and q is the number of cluster centers. Then for a descriptor $\mathbf{h}\in R^d$, LLC finds the nearest k neighbors (K-NN) in $\boldsymbol{B}$ to build a small dictionary $\tilde{\boldsymbol{B}}=[\boldsymbol{b}_{[1]},\cdots,\boldsymbol{b}_{[k]}]\in R^{d\times k}$, where k is a small number that should be pre-defined by users, and the subscript $[k]$ is the index of the kth nearest base in $\boldsymbol{B}$. Finally it uses $\tilde{\boldsymbol{B}}$ instead of $\boldsymbol{B}$ to compute the codes $\mathbf{c}\in R^k$ by optimizing

$$\min_{\mathbf{c}}\|\boldsymbol{h}-\tilde{\boldsymbol{B}}\mathbf{c}\|^2 \quad s.t. \quad 1^T\mathbf{c}=1 \tag{1}$$

where $\mathbf{1}$ is a vector with all ones. Finding the nearest k neighbors actually performs local bases selection, and solving (1) projects every histogram to its local coordinate system, which reserves the locality.

The procedure above is the approximate method of LLC. However it can get almost the same image classification performance as the accurate method [7]. In this paper, we just consider the approximate approach because it is simple and easier to understand.

2.2. Histogram Intersection Kernel

Histogram Intersection Kernel (HIK) was first introduced by [16]. Let $\mathbf{h}=[h_1,\cdots,h_d]^T\in R_+^d$ be a histogram (e.g. HOG feature descriptor), and the Intersection kernel of two histograms $\boldsymbol{h}^{(1)},\boldsymbol{h}^{(2)}$ is defined as follows.

$$K\left(\boldsymbol{h}^{(1)},\boldsymbol{h}^{(2)}\right)=\sum_{i=1}^{d}\min\left(h_i^{(1)},h_i^{(2)}\right) \tag{2}$$

It is proved that this kernel function is a positive definite kernel [22]. According to Mercer's theory, there exists a mapping function $\Phi(\cdot)$, which satisfies

$$K(\boldsymbol{x},\boldsymbol{y})=\Phi(\boldsymbol{x})^T\Phi(\boldsymbol{y}) \tag{3}$$

where $\Phi(\boldsymbol{x})^T$ is the transpose of $\Phi(\boldsymbol{x})$. Through the non-linear mapping $\Phi(\boldsymbol{x})$, histogram similarity is equivalent to an inner product on the feature space. Even $\Phi(\cdot)$ is possibly infinite dimension, we can compute the similarity by kernel trick. This kernel trick makes it possible to use HIK in creating dictionary [14]. More details about dictionary learning will be presented in Section 3.1.

3. A robust local sparse coding method

To combine HIK with LLC, we first create the dictionary $\boldsymbol{B}_\Phi$ on the mapping space. Then for a histogram $\boldsymbol{h}_*$, we map $\boldsymbol{h}_*$ to $\Phi(\boldsymbol{h}_*)$, find the k nearest bases in $\boldsymbol{B}_\Phi$ to build a small dictionary $\tilde{\boldsymbol{B}}_\Phi$. Finally, we optimize the cost function involving $\tilde{\boldsymbol{B}}_\Phi$ to get the codes. In the following, we will detail the above steps.

3.1. Dictionary learning with HIK

The work [15] first introduces HIK into K-Means to learn the visual dictionary, and histograms are compared using HIK instead of l_2 distance. In our method, we reference [15] method with one difference. Algorithm 1 shows the kernel K-Means method.

Algorithm 1. Algorithm for dictionary learning with HIK.

Input A training histogram set $H=(\boldsymbol{h}_1,\ldots,\boldsymbol{h}_N)\in R^{d\times N}$ with N histograms, q (the number of cluster centers), and error threshold ε

Output The dictionary $\boldsymbol{B}_\Phi$

1. $t=0,\mathrm{err}^{(t)}=\infty$
2. Run K-Means++ [23] on the histogram set to choose q histograms $\overline{\boldsymbol{h}}_1,\ldots,\overline{\boldsymbol{h}}_q$, and then take $\boldsymbol{m}_i=\Phi\left(\overline{\boldsymbol{h}}_i\right)$ $(\forall i\in[1,\cdots q])$ as the initial centers on the mapping space, where Φ is the mapping corresponding to the kernel.
3. **Repeat**
4. **For** all $1\le i\le N, l_i=argmin_{1\le j\le q}\|\Phi(\boldsymbol{h}_i)-\boldsymbol{m}_j\|^2$ (assign every data point to one cluster center)
5. **For** all $1\le i\le q$, $\pi_i=j|l_j=i, 1\le j\le N$ (collect the data points for every cluster center)

[1] The dictionary is equivalent to the cluster centers in our paper, and one base in the dictionary stands for one cluster center.

Download English Version:

https://daneshyari.com/en/article/405939

Download Persian Version:

https://daneshyari.com/article/405939

Daneshyari.com