Neurocomputing 172 (2016) 207–214

Contents lists available at ScienceDirect

Neurocomputing

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

Kernelized sparse hashing for scalable image retrieval

Yin Zhang, Weiming Lu *, Yang Liu, Fei Wu

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 30 October 2013
Received in revised form
3 February 2015
Accepted 7 February 2015
Available online 13 May 2015

Keywords:
Image retrieval
Hashing
Sparse coding
Kernel methods

ABSTRACT

Recently, hashing has been widely applied to large scale image retrieval applications due to its appealing query speed and low storage cost. The key idea of hashing is to learn a hash function that maps high dimensional data into compact binary codes while preserving the similarity structure in the original feature space. In this paper, we propose a new method called the Kernelized Sparse Hashing, which generates sparse hash codes with ℓ_1 and non-negative regularizations. Compared to traditional hashing methods, our method only activates a small number of relevant bits on the hash code and hence provides a more compact and interpretable representation of data. Moreover, the kernel trick is introduced to capture the nonlinear similarity of features, and the local geometrical structure of data is explicitly considered in our method to improve the retrieval accuracy. Extensive experiments on three large-scale image datasets demonstrate the superior performance of our proposed method over the examined state-of-the-art techniques.

1. Introduction

With the explosive growth of online images and videos, much effort has been devoted to developing the data-dependent hashing methods that aim to learn the similarity-preserving compact binary codes for representing high-dimensional visual contents [1–11]. This type of hashing technique features two common traits: the encoded data are compact enough to be loaded into the main memory, and the computation of Hamming distance can be efficiently conducted using the bit XOR operation in CPU. As a result, it is very fast to perform approximate similarity computation over such binary codes, which can significantly scale up the nearest neighbor search to large-scale image collections.

One common paradigm of the aforementioned hashing methods is to map the input data into a low-dimensional space by some kinds of embedding techniques, e.g. principal component analysis (PCA), and then binarize them into hash codes. As a result, there are two feasible paths to further boost the performance of learning-based hashing methods: one is to introduce a more advanced dimension reduction technique to obtain the more informative embedded representation of the data [1,3,5,8]. The other is to develop smarter quantization schemes for transforming embedded representations into binary codes [9,12–14]. In this paper, we choose to go down the former path and propose a *Kernelized Sparse Hashing* (KSpH) method, inspired by recent advances in sparse dictionary learning [6,8,15] that bring more flexibility to adapt embedded representations to the data than principal components.

Fig. 1 shows the main process of dictionary learning, code generation and binarization in our framework. Our proposed KSpH method learns the hash codes that minimize the reconstruction error on the pre-trained dictionary. For each data point, our KSpH has the power to activate the most relevant bits to 1 and others to 0 by imposing the *sparsity* and *non-negative* constraints on the reconstruction coefficients. This characteristic enables all hashing bits to be fully utilized since each hashing bit only needs to be effective for certain data points. Furthermore, the generated sparse and non-negative coefficients can be naturally mapped into binary codes without losing too much information. At last, our method is also inspired by previous works [3,7,16,17], and capable of exploiting non-linear kernel function to measure similarity of features and explicitly preserving the local geometrical structure of data, which lead to yielding better hash codes for image retrieval.

The main contributions of this paper are as follows:

- By generating the sparse hash codes with both ℓ_1 and non-negative regularizations, only a small number of relevant hashing bits are activated, which provides a more compact and interpretable representation of data.
- The *kernel trick* is introduced to capture the nonlinear similarity of features, which makes our method more adaptable to different data distributions and non-linear similarity metrics.
- The local geometrical structure of data is explicitly utilized when learning the dictionary, which further improves the performance of nearest-neighbor retrieval.

* Corresponding author.
E-mail addresses: zhangyin98@zju.edu.cn (Y. Zhang), luwm@zju.edu.cn (W. Lu), liuy@zju.edu.cn (Y. Liu), wufei@zju.edu.cn (F. Wu).

http://dx.doi.org/10.1016/j.neucom.2015.02.080
0925-2312/

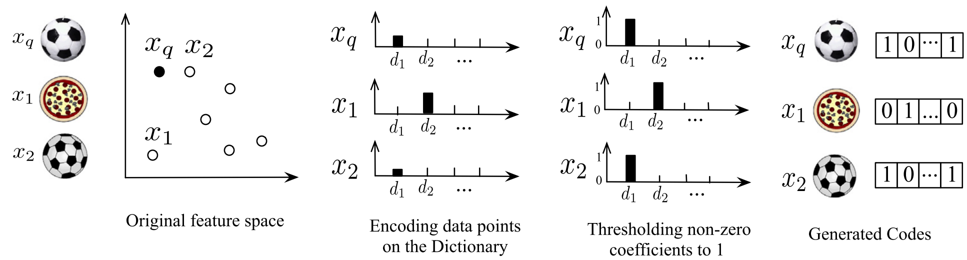

Fig. 1. Image x_q, x_1 and x_2 are firstly represented in their original space. Then the three images are encoded by the dictionary bases learnt from the training data. Finally, non-zero coefficients w.r.t. each data point are binarized into 1.

- We conduct experiments on NUS-WIDE, ANN-GIST and CIFAR dataset which consist of 270,000, 1 million, and 60,000 images, respectively. The experimental results show that the proposed method outperforms the tested state-of-the-art techniques.

The remainder of this paper is organized as follows: Section 2 briefly introduces the related work on hashing methods for scalable image retrieval. Section 3 presents the formal formulation of our proposed KSpH framework. Section 4 shows the details of the alternating optimization algorithm for learning the dictionary and sparse coefficients. Section 5 provides the experiment setup and results. Finally, we draw our conclusions in Section 6.

2. Related work

Faster nearest neighbor search technologies with sub-linear or even constant time complexity are highly desirable for scalable image retrieval and processing applications [1,7,9,10,18,19]. A promising way to expedite nearest neighbor search is to hash the high dimensional data into compact binary codes under the condition of similar data points being mapped into similar codewords within a small Hamming distance. Locality Sensitive Hashing (LSH) [20] is a representative hashing method that uses simple random projections as hash functions. By selecting the hash functions that satisfy the locality sensitive property, similar data will have a high probability to be mapped into the same hash code in LSH. In [2], KLSH is proposed to generalize LSH with kernel functions which makes LSH more flexible in the setting that the embedded feature space is unknown or incomputable. However, a common weakness of LSH and its extensions is that a large number of hash tables are needed to attain a good search performance due to the randomness of hash functions.

To better exploit the data distribution, a plenty of data-aware hashing methods have been proposed by means of machine learning technologies. In [4], stacked Restricted Boltzmann Machines (RBMs) are utilized to capture the data distribution and generate compact binary codes to represent the data points. Spectral Hashing (SH) [1] uses a separable Laplacian eigenfunction formulation that ends up assigning more bits to higher-variance PCA directions. Binary reconstruction embedding (BRE) [3] learns the hash function by explicitly minimizing the reconstruction error between the original distances and the Hamming distances of the corresponding binary codes. In [17], Self-Taught Hashing (STH) is proposed to preserve the local similarity of data and learn the hash function in a self-taught manner. Spline Regression Hashing (SRH) [7] simultaneously considers the local and global similarities of data by combining the local spline functions and the global kernel hashing function.

After obtaining the embedded representations, smart quantization schemes [9,12–14] are able to further improve the quality of final compact binary codes. In iterative quantization [9], learning a good binary code is formulated as the problem of directly minimizing the quantization error of mapping the PCA-projected data to vertices of the binary hypercube. In [12], Manhattan hashing (MH) is proposed to solve the problem of Hamming distance based hashing. The basic idea of MH is to encode each projected dimension with multiple bits of natural binary code (NBC). In [13,14], adaptive multi-bit quantization methods are proposed to quantize each projected dimension with variable bit numbers. More bits will be adaptively allocated to encode dimensions with larger dispersion while fewer bits for dimensions with smaller dispersion.

Modeling data vectors as sparse linear combinations of basis elements [6,8,10,11,15] is widely used in machine learning, signal processing and so forth. Dictionary Learning [15] is focused on learning the basis set to adapt it to specific data. The novel data will be represented with a sparse linear combination of a few atoms in the learned dictionary. Robust Sparse Hashing (RSH) [8] thinks that the input vectors themselves are perturbed or uncertain, and learns dictionaries on the robustified counterparts of uncertain data points. The difference between RSH and our method is that RSH models data uncertainty from a robust optimization perspective, however, our method exploits the local geometrical structure of data and the kernel trick to model complex data distributions.

3. The proposed framework

Given a collection of d-dimensional n data points $X=[x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$, hashing aims to learn a hash function that maps these data points into l-dimensional binary codes $Y=[y_1, y_2, \ldots, y_n] \in 0,1^{l \times n}$. Here, d is the dimensionality of the original feature space and l is the length of the obtained hash codes.

As mentioned above, we expect each hash code y_i to be the sparse representation of its corresponding data point x_i under the given dictionary bases $D=[d_1, d_2, \ldots d_l] \in \mathbb{R}^{d \times l}$, while minimizing the reconstruction error. Inspired by the sparse coding and dictionary learning models [6,15], we write our objective function as follows:

$$\begin{aligned} \underset{\{y_i\}_{i=1}^n, D}{\text{argmin}} : \quad & \sum_{i=1}^{n} \left(\| x_i - D y_i \|_2^2 + \lambda \| y_i \|_1 \right) \\ \text{s.t.} : \quad & y_i \in 0,1^l, \quad i = 1, \ldots, n \\ & \| d_j \|_2^2 \le c, \quad j = 1, \ldots, l \end{aligned} \tag{1}$$

The constraint on y_i enforces the learned hash codes to be binary and the constraint $\| d_j \|_2^2 \le c$ removes the scaling ambiguity.

However, the objective function in Eq. (1) can be demonstrated to be NP hard. Following [1,7], we remove the constraint $y_i \in 0,1^l$ to make the problem computationally tractable. It should be noted that when y_j is enforced to be binary, only dictionary bases that are positively correlated with x_i have the chance to be activated to 1. Thus, we add the non-negative constraint to the relaxed coefficients to eliminate the negatively correlated bases, which keeps