Neurocomputing 151 (2015) 1423–1429

Contents lists available at ScienceDirect

Neurocomputing

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

Two-phase mapping hashing

Wing W.Y. Ng*, Yueming Lv, Daniel S. Yeung, Patrick P.K. Chan

Machine Learning and Cybernetics Research Center, School of Computer Science and Engineering, South China University of Technology, China

ARTICLE INFO

Article history:
Received 23 March 2014
Received in revised form
29 August 2014
Accepted 15 October 2014
Communicated by D. Wang
Available online 29 October 2014

Keywords:
Hashing
Image retrieval

ABSTRACT

Hashing methods have become important retrieval methods for large image databases due to its fast searching time. However, current hashing methods mapping high dimensional real-valued vectors of images to low dimensional hash codes directly may not yield good performance. These methods perform dimensionality reduction together with the Hamming quantization which lead to a difficult nonlinear problem. In this work, we propose a Two-phase Mapping Hashing (TMH) method which first maps images from the high dimensional real-valued space to a high dimensional Hamming space (hash code) using the SKLSH to preserve pairwise similarity. Then, the mapping from the high dimensional Hamming space to a low dimensional Hamming space is found via a minimization of reconstruction error between the two Hamming spaces. The shorter hash code created by the TMH preserves pairwise similarity of images in the input space and yields better precision and recall rates in comparison to SKLSH. Experimental results show that the TMH outperforms the LSH, the Compressed Hashing, the ITQ and the SKLSH.

1. Introduction

Today the image traffic over the Internet increases beyond any bound. For instance, the Facebook receives uploads of more than 2 terabytes of photo every day and it peaks at over 300,000 images per second. Image databases have grown to such a size that linear time search methods are no longer fast enough for content based image retrieval. Sublinear time techniques like hashing based approximated nearest neighbor search attract a lot of attentions from the image retrieval research community [1,2].

Many hashing methods have been proposed in recent years. They can be divided into two major categories: random projection based and data-dependent based methods. The Locality Sensitive Hashing (LSH) is a very first hashing method with a strong theoretical foundation [3–7]. The major drawback of LSH and its variants (e.g. Locality-Sensitive Binary Codes from Shift-Invariant Kernels (SKLSH)) is the requirement of a long hash code to achieve high precision in the similarity preservation. It also results in a poor recall rate. For example, the SKLSH preserves pairwise similarities of images in the input space with a probability of at least $1-N^2e^{-2n\delta^2}$, where N, n and δ denote the number of images in the database, the number of hash bits and a constant [3]. Therefore, a 256-bit SKLSH hash table is $\left(e^{-512\delta^2}\right)^{-3}$ times higher in probability of failing to preserve pairwise similarities in the input space when compared with a 1024-bit SKLSH hash table. To deal with this problem, data-dependent based methods find compact hash codes by mapping from high dimensional real-valued vectors which describe images to low dimensional hash codes. Data-dependent methods achieve satisfying performances with short hash code, but they fail to improve performances when increasing the number of hash bits, as in the case of the LSH. The single mapping function learned by data-dependent methods performs both dimensionality reduction and Hamming quantization in a single step. This is a difficult task because of the highly nonlinear relationship between the high dimensional real-valued space and the low dimensional Hamming space.

To address this problem, we propose a two-phase Two-phase Mapping Hashing (TMH) method to perform the two tasks separately. Fig. 1 shows the basic idea of the TMH. We first map the high dimensional real-valued vectors describing images onto a high dimensional hash code by the SKLSH. Then, in Phase 2, the mapping between the high dimensional hash code and the low dimensional hash code is found via a minimization of reconstruction error between them. The hash code of the SKLSH preserves the similarity between images in the real-valued input space. With this similarity preserving property, the Phase 2 compresses the Hamming space into a lower dimensionality one with minimum reconstruction error. Therefore, the TMH preserves the actual similarity of images with minimum loss in the final low dimensional hash code.

The rest of this paper is organized as follows. Section 2 briefly introduces related works and Section 3 shows the detail of the TMH method. Experimental results are presented and discussed in Section 4. We conclude this work in Section 5.

* Corresponding author.
E-mail address: wingng@ieee.org (W.W.Y. Ng).

http://dx.doi.org/10.1016/j.neucom.2014.10.046
0925-2312/

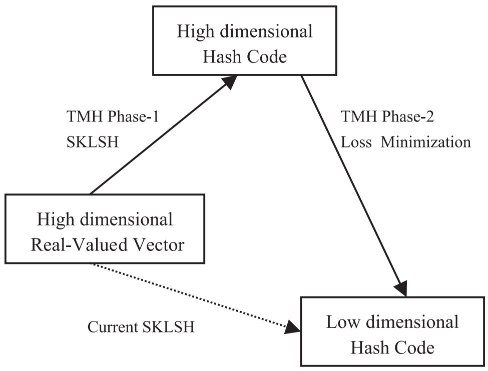

Fig. 1. Mapping schemes of TMH and SKLSH.

2. Related work

LSH is the most widely used random projection based hashing method [5], Variants of the LSH include l_p norm LSH [7], LSH forest [8], entropy-based LSH [9] and multi-probe LSH [10]. There are several kernel projection based LSH methods: Kernel LSH [4], SKLSH [3], and image kernel LSH [11]. The locality preserving principle states that two images will be hashed into the same hash code with a probability directly proportional to the actual similarity between these two images if hash functions are randomly selected. However, all LSH based methods suffer from the problem of requiring a long hash code or a large number of hash tables for a high performance.

Data-dependent hashing methods are proposed to address this problem. Spectral Hashing [12] assumes the data following a uniform distribution and the hash code is computed using a 1D-Laplacian eigenfunction. Unsupervised Sequential Projection Learning for Hashing [8] iteratively learns a hash bit by correcting error made by the previous one. The Iterative Quantization (ITQ) [13,14] minimizes the average quantization error of the hash code with respect to currently assigned hash codes to all images by iteratively rotating the orthogonal projection of hash functions. The quantization error is defined as the difference between the real-valued outputs of hash functions and the binary hash codes. The ITQ uses the top m principal components of PCA on the data as the initial hash functions, where m denotes the number of hash bits. Although the ITQ is one of the best hashing methods, the rotation in the ITQ is to minimize the quantization error instead of the differences in similarity between the actual similarity and the one in the Hamming space. In [15], a boosting based method is applied to the ITQ to create multiple hash tables for image retrieval. Then, a query-adaptive re-ranking is applied to the retrieval results from the multiple tables of the ITQ for a better precision rate. Anchor Graph Hashing [16] obtains a tractable low-rank adjacency matrix using an anchor graph which captures neighborhood structure information in the data. The hash code is then obtained via a Laplacian eigenfunction of the adjacency matrix. Performances of these data-dependent methods will not improve when the number of hash bit increases.

The Compressed Hashing [17] tries to relieve this problem by using a sparse coding and compressed sensing. It first finds sparse real-valued vectors using anchor graph and then projects data to a low dimensional space by a random projection. Finally, hash codes are found by thresholding at the median values of each dimension of the random projection. It preserves the pairwise Euclidean distance in the original sparse vector space based on the Restricted Isometry Property (RIP). The major drawback of the Compressed Hashing is that the RIP only guarantees the similarity preserving between any pairs in the real-valued input space and the randomly projected low dimensional real-valued space, not the binary code. Cartesian k-means [2] is another clustering based hashing method which uses vertices of a binary hypercube as centers and learns the hashing (clustering) using an iterative quantization method similar to the ITQ's. On the other hand, the Angular Quantization-based Hashing [18] focuses on the cosine similarity of the high dimensional non-negative data and creates long codes. Binary reconstructive embedding [19] generates hash functions by minimizing the reconstruction error between the original distances and the Hamming distance of the binary embedding. Minimal loss hashing [20] learns hash functions by minimizing the upper bound of a hinge-like loss function. Kernel based supervised hashing [21] constructs hash functions by minimizing the difference between the normalized inner product of binary codes and pairwise similarity labels of images. Bilinear projection-based hashing [22] focuses on improving the performance of very high dimensional codes for extremely high dimensional visual descriptors (e.g. >4000 bits). It employs a bilinear rotation to maximize the angle between a rotated feature vector and its corresponding binary codes. To reduce the search time used by a long hash code, the Multi-Index Hashing [23] is proposed to divide the long hash code into several disjoint short hash codes to form multiple hash tables. Its search is performed by a method based on the Pigeonhole principle on the multiple hash tables to prevent computing the Hamming distance using all hash bits directly with a linear scan.

Therefore, we propose the TMH to find a low dimensional Hamming space which preserves pairwise similarities of images in the original input space.

3. Two-phase mapping hashing

The two phases of the TMH will be introduced in Sections 3.1 and 3.2, respectively. The computational time complexity is analyzed in Section 3.3. The overall algorithm of the TMH is shown in Fig. 2.

3.1. Phase 1 of TMH

In this phase, the TMH uses the SKLSH to find a high dimensional hash code based on Random Fourier Features (RFFs) [24]. The pairwise similarity between images x and y is defined as $K(x,y)$.

For a mercer kernel $K(\bullet,\bullet)$ which satisfies the following properties P1 and P2:

P1 (Shift invariant): $K(x,y)=K(c+x,c+y)=K(x-y)$

P2 (Normalized): $K(x,y)\le 1$ *and* $K(x-x)=K(0)=1$ where c is a real-valued constant. Let $\Phi_{w,b}=\sqrt{2}\cos\left(w^T x+b\right)$ Then $E(\Phi_{w,b}(x),\Phi_{w,b}(y))=K(x-y)$, where $b\sim Uniform[0,2\pi]$, $w\sim P_k$ and P_k denotes the distribution induced by the kernel. For a Gaussian kernel, $K(x-y)=e^{-\gamma\|x-y\|^2/2}$ and $w\sim P_k\sim Normal\left(0,\gamma I_{D\times D}\right)$. By randomly drawing an i.i.d. sample $((w_1,b_1),(w_2,b_2),\dots,(w_n,b_n))$, we have the following mapping:

$$\Phi^n(\bullet):R^D\to R^n \tag{1}$$

where $\Phi^n(\bullet)=1/\sqrt{n}(\Phi_{w_1,b_1}(\bullet),\Phi_{w_2,b_2}(\bullet),\dots,\Phi_{w_n,b_n}(\bullet))$, D and n denote the number of dimensions of the original input space and the number of dimensions of the mapped space, respectively. Then, $E\left(\Phi^n(x),\Phi^n(y)\right)=K(x-y)$ for all x and y. By the uniform convergence property of the RFF, the new n-dimensional space preserves pairwise similarity of all pairs of images in the original D-dimensional space. Raginsky and Lazebnik [3] prove that by using $F(x)=\left(1+\operatorname{sign}\left(\cos\left(w^T x+b\right)+t\right)\right)/2$ as the mapping function, one could map the RFFs to a random binary code with the following inequality holding true with a high probability for any pair of x and y,