ARTICLE IN PRESS
JID: NEUCOM [m5G;December 14, 2017;20:43]

Neurocomputing 000 (2017) 1–10

Contents lists available at ScienceDirect

Neurocomputing

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

Learning binary codes with local and inner data structure

Shiyuan He[a,1,*], Guo Ye[a,1,*], Mengqiu Hu[a], Yang Yang[a,*], Fumin Shen[a], Heng Tao Shen[a], Xuelong Li[b]

[a] Center for Future Media and School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, PR China
[b] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi 710119, PR China

ARTICLE INFO

Article history:
Received 23 August 2017
Revised 23 November 2017
Accepted 2 December 2017
Available online xxx

Communicated by Xiaofeng Zhu

Keywords:
Supervised hashing
Anchor graph
Nearest neighbor search

ABSTRACT

Recent years have witnessed the promising capacity of hashing techniques in tackling nearest neighbor search because of the high efficiency in storage and retrieval. Data-independent approaches (e.g., Locality Sensitive Hashing) normally construct hash functions using random projections, which neglect intrinsic data properties. To compensate this drawback, learning-based approaches propose to explore local data structure and/or supervised information for boosting hashing performance. However, due to the construction of Laplacian matrix, existing methods usually suffer from the unaffordable training cost. In this paper, we propose a novel supervised hashing scheme, which has the merits of (1) exploring the inherent neighborhoods of samples; (2) significantly saving training cost confronted with massive training data by employing approximate anchor graph; as well as (3) preserving semantic similarity by leveraging pair-wise supervised knowledge. Besides, we integrate discrete constraint to significantly eliminate accumulated errors in learning reliable hash codes and hash functions. We devise an alternative algorithm to efficiently solve the optimization problem. Extensive experiments on various image datasets demonstrate that our proposed method is superior to the state-of-the-arts.

1. Introduction

During the last decades, visual search for large-scale dataset has drawn much attention in many content-based multimedia applications [1–3]. Among in large-scale vision problems, binary coding has attracted growing attention in image retrieval [4–6], image classification [7,8] and etc. Especially in recent years, the demand for retrieving relevant content among massive images is stronger than ever under such a big-data era. For most occasions, users input a keyword and intend to obtain the semantic similar images precisely and efficiently. For servers, it is almost impossible to linearly search the objects especially when they confront vast amount of images (e.g., the photo sharing website Flickr possesses more than five billion images and people still upload the photos at the rate of over 3000 images per minute) [9]. In contrast to the cost of the storage, the search task consumes more computational resources due to the massive search request. Therefore, recent years people devoted a lot of efforts to handle this problem. Among these proposed methods, hash shows great superiority than others. In simple terms, hashing methods map high-dimensional images into short binary codes and preserve the similarity of the original data at the same time. In this way, searching similar images converts to finding neighbor hashing codes in Hamming space which is simple and practical. Consequently, the technique leads to significant efficiency in multimedia, computer vision, information retrieval, machine learning and pattern matching [10–19].

For many applications, approximate nearest neighbor (ANN) search is sufficient enough [20–24]. Given a query instance, the algorithm aims to find the similar instance instead of returning the exact nearest neighbor. Therefore, efficient data structure is required to store data for fast search. Under such background, tree-based indexing approaches are proposed for approximate nearest neighbor (ANN) search typically with the sub-linear complexity of $O(log(N))$. Among them, KD tree [25–27], R tree [28] and metric tree [29] are most representative. However, as the image technology develops, the descriptors of one image usually reach to hundreds dimensions and with the dimension increasing, the tree-based methods' performance dramatically degrades and it need more space to store data which costs a lot. In consideration of the inefficiency of tree-based indexing methods, hashing approaches have been proposed to map entire dataset into discrete codes and

* Corresponding authors.
E-mail addresses: shiyuanhe.david@gmail.com (S. He), michaelyeah7@gmail.com (G. Ye), hmq.uestc@gmail.com (M. Hu), dlyyang@gmail.com (Y. Yang), fumin.shen@gmail.com (F. Shen), shenhengtao@hotmail.com (H.T. Shen), xuelong_li@opt.ac.cn (X. Li).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.neucom.2017.12.005
0925-2312/© 2017 Published by Elsevier B.V.

preserve the similarity of the data at the same time. The similarity between data points can be measured by Hamming distance which costs little time to calculate for computers.

The existing hash methods can be roughly divided into two groups [30]: data-independent and data-dependent. One of the most classic data-independent methods is Locality-Sensitive Hashing (LSH) [31] that has been widely used to handle massive data due to its simplicity. LSH uses a hash function that randomly projects or permutates nearby data points into same binary codes. However, LSH needs long binary codes to achieve promising retrieval performance which increases the storage space and computation costs. Moreover, LSH ignores the underlying distributions and manifold structure of the data on account of random-projection.

Realizing this deficiency, Weiss et al. proposed Spectral Hashing (SH) [32] utilizing the subset of thresholded eigenvectors of the graph Laplacian by relaxing the original problem which improves the retrieval accuracy to some extent. Yet it charges more time to build a neighborhood graph. Liu et al. delivered some improvements to SH and proposed the Anchor Graph Hashing (AGH) [33] that uses anchor graphs to obtain low-rank adjacency matrices. Formulation of AGH costs constant time by extrapolating graph Laplacian eigenvectors to eigenfunctions. Note that SH and AGH is data-dependent cause it exploits the feature information of the data and preserve the metric structure. This kind of methods are called unsupervised methods like principal component analysis based hashing (PCAH) [34], iterative quantization (ITQ) [35], isotropic hashing (Iso-Hash) [36] and an affinity-preserving k-means hashing (KMH) [37].

However, hashing methods mentioned above can not achieve high retrieval performance with a simple approximate affinity matrix [38]. Due to the semantic gap where the visual similarity often differs from semantic similarity, returning the nearest neighbors in metric space can not guarantee search quality [34,39]. To solve this problem, images that are artificially labeled as similar or dissimilar are used by supervised hashing methods [5,40–47]. The following are most popular ones among them. Kernel-Based Supervised Hash (KSH) [48] maps data into Hamming space where similar items have minimal Hamming distance and simultaneously dissimilar items have maximum distance. Binary reconstructive embeddings (BRE) [49] proposed the hash function which can minimize the reconstruction error between the metric space and Hamming space. Canonical Correlation Analysis with Iterative Quantization (CCA-ITQ) [35] and Supervised Discrete Hashing (SDH) [50] are proposed to satisfy the semantic similarity. By leveraging pairwise labeled information, the performance of supervised methods has been remarkably improved. Moreover, some hashing learning approaches based deep neural networks have been recently proposed to perform simultaneous.

No matter the method is supervised or unsupervised, the object function with discrete constraints involves a mixed binary-integer problem [50] which is NP-hard. To tackle this problem, most hash methods relax the discrete constraints. They first calculate a continuous solution then threshold it to obtain binary codes without realizing the importance of discrete optimization. This technique will leads to the significant information loss during the learning process [51]. It has been shown that the quality of the codes degrades quickly especially when the code length increases if we ignore the discrete constraints. Some methods try to improve the quality by replacing the sign function with more smoothing sigmoid function [52]. However, it does not solve the problem explicitly. Recently, only few methods directly generate hash codes in discrete space. In addition, deep neural network has been applied in recent retrieval progress [53–55]. Deep-learning-based hashing obtains high efficiency by learning the image representation and hash codes tightly coupled [56,57]. Nonetheless, due to the resulting computational expense and storage cost are huge, we do not make much comparison.

In this paper, we aim to design a supervised hash method which can efficiently generate high-quality compact codes. We utilize the anchor graph which is built based on the pairwise similarity to exploit the inner structure of the original data, in the process of learning hash function, we also take the supervision information to preserve the pairwise similarity to improve the accuracy of retrieval. To avoid accumulated errors caused by continuous relaxation, we choose to directly optimize the binary codes. With the discrete constraints added to objective function, we propose a novel hashing framework, termed Local and Inner Data Structure Supervised Hashing (LISH) which is able to efficiently generate codes and satisfy the semantic similarity at the same time. We enhance our work by conduct more experiments to assure the results much more detailed and accurate [58]. We also simplify some of our mathematical derivation. Our main contributions are summarized as follows:

- Our method uses graph Laplacian to captures the local neighborhoods to enhance hashing codes' quality. And semantic gap can be properly solved by utilizing labeled information. By this way, both metric and semantic similarity are preserved by our method which contribute a lot to improve the performance significantly.
- Most existing hash method solve the problem with the relaxation of the discrete constraints, since we directly optimize our method and each bit can be sequentially learned by the algorithm, our method outperforms in an alternative and efficient manner.
- We evaluate our method on three popular large-scale image datasets and obtain superior accuracy than state-of-the-arts.

The remainder of this paper is organized as follows. A brief review of related work is in Section 2. We present the detailed formulation of proposed LISH method in Section 3. Experimental results are given in Section 4. We conclude our work in Section 5.

2. Related work

Suppose we have n samples $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, deposited in matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n\times d}$, where d is the dimensionality of the feature space. In the this section, we briefly review a few representative methods including Spectral Hashing (SH), Kernel Supervised Hashing (KSH).

2.1. Spectral hashing

Spectral Hashing is one of the most popular data-dependent hash methods. It generates bits by spectral graph partition [59]. Specifically, it thresholds the eigenvectors of Laplacian graph. it maps similar data points from input space into Hamming space and preserve their similarity at the same time. The object function of Spectral Hashing is formulated as follows:

$$\begin{aligned} &\min \sum_{ij} \mathbf{a}_{ij} \|\mathbf{b}_i - \mathbf{b}_j\|^2 \\ &subject\ to:\ _{ij} \in -1, 1^k,\ \sum_i \mathbf{b}_i = 0,\ \frac{1}{n}\sum \mathbf{b}_i\mathbf{b}_i^T = I, \end{aligned} \tag{1}$$

where $\mathbf{a}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\epsilon^2)$ represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{b}_i$ is the corresponding binary codes of $\mathbf{x}_i$, the radius parameter ϵ defines the distance which corresponds to similar items. It is obvious that the more similar $\mathbf{x}_i$ and $\mathbf{x}_j$ are, the value of the $\mathbf{A}_{ij}$ becomes larger, which leads to the smaller Hamming distance $\|\mathbf{b}_i - \mathbf{b}_j\|^2$. The constraint $\sum_i \mathbf{b}_i = 0$ ensures the hash bit to be balanced in order to maximize the information. The constraint