Neurocomputing 178 (2016) 112–122

Contents lists available at ScienceDirect

Neurocomputing

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

A fast binary encoding mechanism for approximate nearest neighbor search

Hongwei Zhao [a,b], Zhen Wang [a], Pingping Liu [a,b,*], Bin Wu [a]

[a] School of Computer Science and Technology, Jilin University, Changchun 130012, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun 130012, China

ARTICLE INFO

Article history:
Received 31 January 2015
Received in revised form
3 August 2015
Accepted 8 September 2015
Available online 11 November 2015

Keywords:
Hashing algorithm
Binary codes
Approximate nearest neighbor search
Image retrieval

ABSTRACT

In this paper, a novel approach which can map high-dimensional, real-valued data into low-dimensional, binary vectors is proposed to achieve fast approximate nearest neighbor (ANN) search. In our paper, the binary codes are required to preserve the relative similarity, which makes the Hamming distances of data pairs approximate their Euclidean distances in ANN search. Under such constraint, the distribution adaptive binary labels are obtained through a lookup-based mechanism. The perpendicular bisector planes located between two kinds of data whose binary labels are different on only one specific bit are considered as weak hash functions. As just two kinds of data are taken into consideration during generation of the weak hash functions, the final strong hash functions are formed by combining the weak ones through boosting scheme to map all kinds of data into binary codes effectively. Experimental results show that our algorithm can encode the out of samples efficiently, and the performances of our method are superior to many state-of-the-art methods.

1. Introduction

Recently, hashing methods have been widely applied in image retrieval [1–4], large-scale object detection [5–7], mobile vision search [8] and so on. With the wide spread of mobile computing application and ever-increasing amount of image data, the real-valued features with high dimension such as SIFT [9] and GIST [10] are no longer appropriate choices for fast image retrieval. Many promising methods have been proposed to fix up such problems.

Hashing algorithms aim to map high-dimensional, real-valued data into compact binary codes, while preserving the locality sensitivity [7] which demands that the Hamming distances of similar data pairs in the Euclidean space should be small enough. The classical method, locality sensitive hashing (LSH) [11], generates random hyper projection planes to map data into binary codes. As the hash functions in LSH are data independent, its performance cannot improve significantly as the number of binary bits increasing [7]. Spectral hashing (SH) [7] obtains binary codes through partitioning spectral graph, and it has better performance for the data with uniform distribution in a high-dimensional rectangle. Liu et al. [12] consider the cluster centers obtained through the k-means method as graph nodes, and then the binary codes are obtained through partitioning the established graph via spectral relaxation. The binary codes obtained in [13] aim to explicitly minimize the reconstruction error between the original distances and the Hamming distances of the corresponding binary embedding. Minimal loss hashing (MLH) [14] sets thresholds in Euclidean space and Hamming space separately. In the MLH method, the relative relationships between the Hamming distances of data pairs and the Hamming threshold should be as the same as that between their Euclidean distances and the Euclidean threshold. A kernel-based supervised hashing model which requires a limited amount of supervised information and a feasible training cost in achieving high quality hashing is proposed in [15], and its main idea is to obtain binary codes whose Hamming distances are minimized on similar data pairs and maximized on dissimilar pairs. ITQ [16] learns binary codes through rotating data to align to the vertices of binary hyper cube, which makes the binary codes that are not distribution adaptive. The distribution adaptive binary codes are obtained in KMH [17] which utilizes k-means like iteration process to obtain encoding centers.

As mentioned by He et al. [17], existing hashing algorithms can be divided roughly into Hamming-based methods [11,3,18,19] and lookup-based methods [17,20,21]. Hamming-based methods partition the original space into different parts using hyper planes [11,22,23,14], and the real-valued data in each part are mapped into unique binary codes according to the signs of the projection results with the hyper planes. Each bit of the binary codes in

* Corresponding author at: School of Computer Science and Technology, Jilin University, Changchun 130012, China.
E-mail addresses: zhaohw@jlu.edu.cn (H. Zhao), wangzhencst@gmail.com (Z. Wang), liupp@jlu.edu.cn (P. Liu), wubin14@mails.jlu.edu.cn (B. Wu).

http://dx.doi.org/10.1016/j.neucom.2015.09.110
0925-2312/

Hamming-based methods is determined by one hyper plane. Lookup-based methods divide the original space into parts with minimal quantization error through the mechanism likes the k-means method [24], which makes the partition results more distribution adaptive than those using hyper planes. After the partition process, the centers of all parts are assigned unique binary codes, and the out of samples are mapped into the same binary codes as their closest centers. Thanks to the adaptive k-means quantization process, the lookup-based methods [21,20] have been shown to be more accurate than some Hamming-based methods [11,19]. However, the time complexity of the encoding process of lookup-based methods is higher. Suppose mapping the data into M-bit binary codes, 2^M encoding centers should be obtained in lookup-based methods, and the unseen data are mapped into the same binary codes as their closest centers according to the relative distances between the unseen data and 2^M centers. Correspondingly, the encoding time complexity of lookup-based methods is $O(2^M)$. In the same case, Hamming-based methods only need M hyper planes to obtain M-bit binary codes. As a result, each data can be encoded just according to M projection results with the encoding time complexity of $O(M)$.

The aim is to improve the encoding efficiency, some hashing algorithms with the two-step mechanism are proposed in [25–28]. Zhang et al. [25] first find the binary codes of training documents, and then regard the linear SVM planes as hash functions. Unseen data are encoded through computing the projection results with classifier planes, so the time complexity of the encoding procedure is acceptable. Lin et al. propose the two-step hashing method in [26], furthermore exploit it in [27]. Lin et al. [27] employ a GraphCut mechanism which is based on the block search method to learn binary codes, and then train boosted decision trees to recompute these binary codes. Zhu et al. [28] obtain the coefficients of sparse items in the first step, in the second step the positive ones are encoded as '1' and the zero coefficients are encoded as '0'.

Besides the encoding efficiency, the other key point is to guarantee the Hamming distances of data pairs can approximate their Euclidean distances in ANN search. Norouzi et al. [14] employ hinge-like loss function to penalize the similar (or dissimilar) data pairs when their Hamming distances are larger (or smaller) than the threshold. The relative similarity defined over triplets of items is used to formulate the hashing problem in triplet loss hashing (TLH) [29]. Wang et al. [30] also employ the triplets of items like in [29] to solve the objective of the listwise loss. Order preserving hashing [31] learns hash functions through maximizing the alignment between the similarity orders in the Euclidean space and those in the Hamming space. Zhang et al. [32] take the topology which represents the neighborhood relationships between subregions and the relative proximities between the neighbors of subregions into consideration during generation of binary codes, and its objective function is optimized through spectral relaxation. Zhao et al. [33] emphasize that the ordinal relationship in the Hamming space should be consistent with that in the Euclidean space. Some hashing methods [14,29,30] just focus on the relative relationships among triplets of items. Fortunately, the objective function in order preserving hashing [31] aims to preserve the relative relationships among all items. However, the bucket balance requirement in [31] is unrealistic for real skewed data.

Hamming-based methods are more attractive in the encoding time complexity compared with lookup-based methods. However, the hash functions of Hamming-based methods are not adaptive enough to data distribution. In this paper, we try to build an effective mechanism which leverages data distribution information to generate hashing projection functions, therefore the out of samples can be mapped into binary codes according to the signs of projection results efficiently. The mechanism which enjoys the merits of both Hamming-based methods and lookup-based methods is proposed to achieve the above challenges, and it is inspired by the hashing algorithms with the two-step mechanism [25–28]. In order to make Hamming distances approximate Euclidean distances in ANN search, the weighted inverse order error which requires the numerical ordinals of data pairs in the Hamming sequence should be consistent with those in the Euclidean sequence is proposed in this paper. The sequences in different spaces are formed through sorting data pairs according to their distance values. Through satisfying the above requirement, the binary codes in our method can well preserve the relative similarity [14,31,29,30,32] which are superior to the absolute distance preserving methods in ANN search [31]. The relative relationships among all items in data set are taken into account in our method. Furthermore, the distribution adaptive mechanism is adopted to fix up the category problem, which makes our method advantage in dealing with real skewed data. The flow chart of our algorithm is shown in Fig. 1. In our paper, the supervision information of the binary labels with minimal weighted inverse order error is first learnt through the unsupervised lookup-based mechanism. The projection planes located between two kinds of data with specific binary labels are considered as the weak hash functions $\{h_k^1, h_k^2, \ldots, h_k^t\}$. Different from the classical LSH method, the hash functions in our method are generated under the supervision of the training samples with binary labels. As a consequence, our hash functions are adaptive to the distribution of data. However, the neighborhood information which emphasizes that the similar data points should be mapped into the same binary codes may be lost during the process of generating the binary labels and the

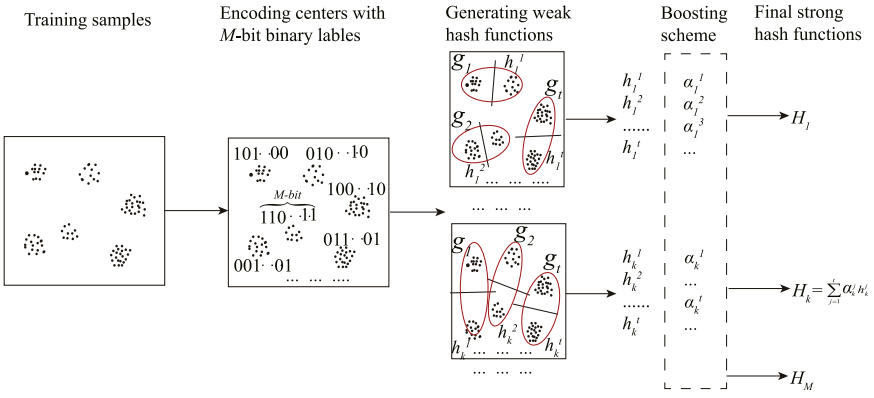

Fig. 1. The flow chart of our algorithm. Firstly, the distribution adaptive binary labels B_i of the data are learnt. Secondly, the weak hash functions are obtained on the basis of the data with specific labels, and they are combined to form the strong ones through boosting scheme.