Pattern Recognition 48 (2015) 3169–3179

Contents lists available at ScienceDirect

Pattern Recognition

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

The ordinal relation preserving binary codes

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

[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 30 September 2014
Received in revised form
26 January 2015
Accepted 16 February 2015
Available online 25 February 2015

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

ABSTRACT

Hashing algorithm has been widely used for efficient approximate nearest neighbor (ANN) search. Learning optimal hashing functions has been given focus and it is still a challenge. This paper aims to effectively and efficiently generate relative similarity preserving binary codes. Most existing hashing methods try to preserve the locality similarity by preserving direct distance similarity, while ignoring the relative similarity which advantages in ANN search. In this paper, this issue is solved by proposing the relative error which emphasizes that the ordinal relations in Hamming space and Euclidean space should be consistent with each other. We learn hashing projection functions via two steps. The first step adopts the lookup-based mechanism to find the optimal binary codes of training data, which can preserve the relative similarity and simultaneously adapt to data distribution. The binary codes in the first step are considered as supervision information in the second step. The objective of the second step is to learn hashing projection functions which can efficiently regenerate the binary codes in the first step. Aim to be in accordance with the property of data distribution, the hyper internal tangent planes of two specified spheres are chosen as hashing projection functions. Assisted by these projection functions, the time complexity of encoding process is greatly reduced. Experimental results on four public data sets demonstrate that our method outperforms many other state-of-the-art methods.

1. Introduction

The widespread of mobile computing applications and increasing amount of image data have rendered floating-point feature descriptors [1,2] with high dimensions inappropriate choices for fast image retrieval. Many novel promising methods, such as sparse coding [3–6] and binary feature learning [7–16] have been proposed to address this challenge.

Sparse coding represents an image as a sparse linear combination of items from an over-complete dictionary. The performance of sparse coding is significantly affected by the quality of the dictionary, and the issue of learning the optimal dictionary has been given focus on. In [3], a unified objective function is formed by combining the "discriminative sparse-code error" with other errors, and a single over-complete dictionary is learned by solving the objective function using the K-SVD algorithm. The graph topology selection problem is adopted in [4] to guarantee the similarity of sparse codes of the feature points belonging to the same class and therefore improve the performance of the dictionary. Inspired by the human visional system, Zhu and Shao [5] introduce a visual categorization framework that utilizes weakly labeled data from other domains as source data to span the intra-class diversity of the original learning system.

Binary feature results in great efficiency gains in storage and facilitate image retrieval using simple data structures and algorithms. Given these outstanding performances, the problem of learning binary features has become a research focus and the methods can be roughly classified into two categories as discussed below.

The first category [7,11,8–10,17] generates binary feature as a concatenation of simple intensity comparison from a raw image patch, which results in an extremely short computation time. Calonder et al. [7] generate binary features using the relative intensity of point pairs and provide five pre-determined spatial arrangements, while ignoring the problem of rotation invariant. The issue of rotation invariant is considered in [8] and the vector from the center of the corner to the centroid is utilized as the main orientation. An exhaustive set of comparisons of close locations is adopted in [10]. Alahi et al. [9] choose the point pair with the highest bit variance. These methods [7–10] generate binary descriptors on the basis of pixel intensity. Trzcinski et al. [11] learn binary descriptors according to the gradient orientation of pixels and adopt an Adaboost-like mechanism to improve the performance. Shao et al. [18] propose to automatically generate domain-adaptive global feature descriptors with multi-objective genetic programming which constitute a suitable mechanism for different image domains.

The second category, hashing method, maps high-dimensional descriptors [1,2] into compact binary codes. The simple and classical

* Corresponding author at: School of Computer Science and Technology, Jilin University, Changchun 130012, China.
E-mail addresses: zhaohw@jlu.edu.cn (H. Zhao), wangzhen14@mails.jlu.edu.cn (Z. Wang), liupp@jlu.edu.cn (P. Liu).

http://dx.doi.org/10.1016/j.patcog.2015.02.011
0031-3203/

method, locality sensitive hashing (LSH) [19], randomly generates hashing projection functions and maps data into binary codes according to mapping results. Given that random hashing functions are data independent, the performance of LSH does not improve obviously as the number of binary bits increases. Torralba et al. [13] introduce binary codes to the vision community and achieve effective results. In [20], the binary codes are generated on the basis of boosting restricted Boltzmann machines. Spectral hashing (SH) [14] learns binary codes by partitioning the spectral graph. However, SH highly demands that the data distribution is uniform which is unrealistic for real data [21]. Gong and Lazebnik [21] rotate the principal component analysis (PCA)-projected data and map the data into the same binary codes as their nearest vertex of a hyper binary cube. Raginsky and Lazebnik [22] find binary codes based on the random feature mapping for shift-invariant kernels.

Hashing method aims to approximate Euclidean distance by Hamming distance. This paper achieves this aim with the help of the relative error, which demands that the relative relationship between the distances of any two data pairs in Hamming space and Euclidean space should be consistent with each other. By satisfying this requirement, this study preserves the relative similarity investigated in [23–27]. Norouzi and Blei [23] employ a hinge-like loss function to punish similar (or dissimilar) points when their hamming distances are larger (or smaller) than the Hamming threshold. Triplet loss hashing [25] uses the relative similarity defined over triplets of items to formulate the hashing problem. Wang et al. [26] also employ the triplets of items like in [25] to solve the objective of the listwise loss. Wang et al. [24] learn hashing functions by maximizing the alignment between the similarity orders in Euclidean space and those in Hamming space. The topology is employed in [27] to preserve the neighborhood relationships and the neighbor ranking jointly, and the learning stage is formulated as a generalized eigen decomposition problem with closed form solutions which is distinct from prior works. These methods [23,25,26] focus on the relative relationships among triplets of items. Our method determines binary codes through the relative relationships among all the items in the dataset. Unrealistic bucket balance restriction is employed to fix the category problem in [24]. By contrast, our method solves the category problem using the distribution adaptive mechanism, which has advantage in dealing with real skewed data.

As discussed in [28], existing hashing methods are roughly classified into Hamming-based methods [19,29,22], lookup-based methods [28,30,31] and others [14,32,33]. Hamming-based methods use hyper planes or kernelized hyper planes to quantize space into cells and map data points into binary codes according to the projection signs. Lookup-based methods usually employ a clustering method to divide data into groups and map the data into the same binary codes as their nearest center. Lookup-based methods have superior performance in adapting to data distribution. However, the encoding time complexity of lookup-based methods is higher than that of Hamming-based methods. Mapping unseen data into m-bit binary codes, lookup-based methods need to compute the distances between the unseen data and 2^m centers, with a time complexity of $O(2^m)$. By contrast, Hamming-based methods only compute m projection results with a time complexity of only $O(m)$.

Attracted by the efficient encoding mechanism of Hamming-based methods, we also try to adopt the projection mechanism to map unseen data into binary codes. Different from LSH [19] which randomly generates its hashing projection functions, in our method, we demand the projection planes should be distribution adaptive as lookup-based methods. In order to achieve this aim, we propose an effective mechanism inspired by two-step hashing algorithms [34–37]. We learn the binary codes of the training samples in the first step and generate hashing functions under the supervision of the binary codes obtained in the first step. The first step in [34] employs spectral hashing [14] to find the optimal binary codes of training documents, and then regard linear support vector machines (SVMs) as hashing projection functions. Unseen data are encoded by computing projection results with classifier planes and the time complexity of the encoding procedure is acceptable. Lin et al. propose a two-step idea in [35] and further exploit this idea in [36]. In [36], a graph cut based block search method is employed to learn binary codes in the first step, and then boosted decision trees are trained to re-compute these binary codes. Sparse hashing [37] generates the coefficients of the sparse items in the first step, and the second step maps the positive ones into "1" and the zero coefficients into "0". The spectral hashing mechanism in [34] demands unrealistic data distribution. To get ride of such requirement, our first step adopts the lookup-based mechanism to compute the binary codes, which makes our binary codes adaptive to data distribution. Our second step considers some specified tangent planes as the projection planes, which avoids computing the complex classification problem in [34,36].

The framework of our method is shown in Fig. 1. In the first step, the relative similarity preserving binary codes are acquired, and we consider these binary codes as supervision information for learning hashing mapping functions in the second step. The first step generates the optimal binary codes using the expectation maximization (EM) algorithm with iterative mechanism, and the objective function is formulated by combining the relative error with the quantization error. The relative error emphasizes preserving relative similarity which benefits the absolute distance preserving methods in approximate nearest neighbor search [24], and the quantization error makes the binary codes adaptive to data distribution. In the second step, we aim to learn hashing projection functions to efficiently regenerate the binary codes in the first step. According to the data with the same binary labels distribute in the same hyper sphere [38], we use the internal tangent planes of two specified spheres as hashing projection functions. Assisted by these hashing projection functions, we can efficiently and effectively calculate the binary codes of the unseen data.

Our main contributions are as follows:

1. A novel hybrid model including two steps is proposed to combine the advantages of lookup-based methods and Hamming-based methods. The first step adopts the lookup-based mechanism to guarantee our hashing functions are adaptive to data distribution, and the second step learns hashing projection functions to greatly reduce the time complexity of the encoding procedure.

2. In the first step, the relative error which aims to preserve relative similarity is proposed to boost the performance of our method. The

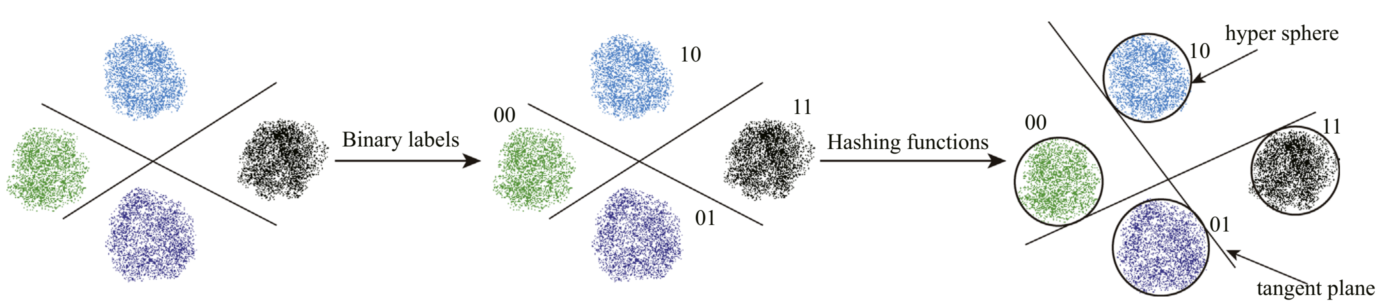

Fig. 1. The framework of our algorithm.