ARTICLE IN PRESS
JID: NEUCOM [m5G;October 2, 2017;15:52]
Neurocomputing 000 (2017) 1–8
Contents lists available at ScienceDirect
Neurocomputing
journal homepage: www.elsevier.com/locate/neucom

Bagging–boosting-based semi-supervised multi-hashing with query-adaptive re-ranking

Wing W.Y. Ng[a], Xiancheng Zhou[a], Xing Tian[a,*], Xizhao Wang[b], Daniel S. Yeung[a]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China

ARTICLE INFO

Article history:
Received 19 January 2017
Revised 8 July 2017
Accepted 13 September 2017
Available online xxx

Communicated by Yongdong Zhang

Keywords:
Semi-supervised information retrieval
Multi-hashing
Bagging
Boosting

ABSTRACT

Hashing-based methods have been widely applied in large scale image retrieval problem due to its high efficiency. In real world applications, it is difficult to require all images in a large database being labeled while unsupervised methods waste information from labeled images. Therefore, semi-supervised hashing methods are proposed to use partially labeled database to train hash functions using both the semantic and the unsupervised information. Multi-hashing methods achieve better precision-recall in comparison to single hashing method. However, current boosting-based multi-hashing methods do not improve performance after a small number of hash tables are created. Therefore, a bagging–boosting-based semi-supervised multi-hashing with query-adaptive re-ranking (BBSHR) is proposed in this paper. In the proposed method, an individual hash table of multi-hashing is trained using the boosting-based BSPLH, such that each hash bit corrects errors made by previous bits. Moreover, we propose a new semi-supervised weighting scheme for the query-adaptive re-ranking. Experimental results show that the proposed method yields better precision and recall rates for given numbers of hash tables and bits.

1. Introduction

The explosive growth of multi-media contents on the Internet creates a huge challenge for image retrieval researches. Image retrieval methods can be categorized into text-based [1–3] and content-based (CBIR) [4–6]. CBIR methods develop rapidly in the past decades. For a large scale CBIR problem, linear search methods may still use too much time and therefore sub-linear methods are needed. Instead of taking a long time to search for exact matches, approximated nearest neighbor search methods [7,8] finding similar images in an approximated manner are much more efficient, especially for very large scale problems and no particular image is targeted. Hashing-based image retrieval methods [9–11] are instances of approximated nearest neighbor search methods which represent images with binary hash codes and have shown to be highly efficient in large scale image searches [12]. For a given query image q, hashing method tries to find its similar by finding images in the database yielding the smallest Hamming distances from q in their hash codes. Therefore, hashing methods generate hash codes for images such that similar images share similar hash codes while dissimilar images have very dissimilar hash codes.

In general, image retrieval performance improves when more hash bits and multiple hash tables are used. However, existing boosting-based multi-hashing methods do not improve and even sometimes reduce retrieval performance after the number of hash tables reaches a certain threshold. Therefore, the bagging–boosting-based semi-supervised multi-hashing method is proposed to address this problem. The proposed method consists of two steps: multi-hashing construction and query-adaptive re-ranking.

Major contributions of this paper include:

- The proposal of a semi-supervised multi-hashing using bagging to relieve the disadvantage of boosting-based multi-hashing methods: new hash table being highly similar to existing one's after a number of tables being created. Then, boosting is used to train individual hash function in each hash table. This hybrid method takes advantages of both bagging and boosting and applies them in different parts of the whole algorithm to maximize their benefits.
- Proposing a semi-supervised weighting scheme for query-adaptive re-ranking to improve retrieval performance of multi-hashing for semi-supervised image retrieval problem.

Related works are introduced in Section 2. The bagging–boosting-based semi-supervised multi-hashing with re-ranking (BBSHR) is proposed in Section 3. Experimental results are shown and discussed in Section 4. Section 5 concludes this paper.

* Corresponding author.
E-mail address: shawntian123@gmail.com (X. Tian).

https://doi.org/10.1016/j.neucom.2017.09.042
0925-2312/

Please cite this article as: W.W.Y. Ng et al., Bagging–boosting-based semi-supervised multi-hashing with query-adaptive re-ranking, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2017.09.042

2. Related works

Current hashing methods and multi-hashing methods are introduced in Sections 2.1 and 2.2, respectively.

2.1. Current hashing methods

Hashing methods can be generally divided into three categories: unsupervised, semi-supervised, and supervised hashing methods according to the usage of semantic information. Unsupervised hashing methods [13–17] do not use semantic information from the given database. The Locality Sensitive Hashing (LSH) and its variants [18,19] are the most representative unsupervised hashing methods which create hash functions by random. The Principle component hashing [20] is another unsupervised hashing method which builds hash functions based on the principle component analysis. The iterative quantization hashing (ITQ) [21] finds binary hash codes for images via a minimization of the quantization error between the real-valued data vectors and the binary hash codes. The objective function of the ITQ is minimized by updating the rotation matrix and hash codes iteratively. The Unsupervised Bilinear Local Hashing applies bilinear projections to generate hash codes [16]. Instead of using hash hyperplanes to divide the data space, the spherical hashing finds hyperspheres to partition the image set into different hash buckets (codes) to generate efficient hash codes [17]. Semantic labels in the database provide extra discriminative information and images with the same class label should have the same or very similar hash codes. The LDAHash [22] and the Supervised Hashing with Kernels [23] are instances of supervised hashing methods which require all images in the database to be labeled and use those labels to learn hash functions. The LDAHash applies the linear discriminant analysis to the data features to build hash functions [22]. The Supervised Hashing with Kernels trains hash functions by maximizing Hamming distances between dissimilar data pairs and minimizing Hamming distances between similar data pairs. Supervised hashing methods usually achieve better retrieval performance in comparison to unsupervised hashing methods. Deep learning is also applied to learn effective hashing by the preservation of the semantic information of images [24]. However, in real world applications, image databases are usually partially labeled and requiring all images being labeled is not practical. Moreover, supervised hashing methods tend to overfit when databases cannot provide enough semantic labels.

Therefore, semi-supervised hashing methods [25–28] are proposed to fully utilize the partial labeled images and the other large portion of unlabeled images. The Sequential Projection Learning for Hashing (SPLH) [25] is one of the representative semi-supervised hashing methods which learns hash bits sequentially and corrects error made by the previously learned hash bit. The Bootstrap Sequential Projection Learning Hashing (BSPLH) is another representative semi-supervised hashing method which learns hash bits sequentially and corrects error made by all previous hash bits [26]. The Deep Learning Hashing [29] generates hash codes of training images by the relative similarity graph and learns hash functions from them using the convolution neural network with both visual and semantic information. In [30], two kinds of contextual query expansions (visual world-level and image level) are proposed based on common visual patterns to improve the performance of image retrieval. The topology preserving hashing trains hash functions incorporating the neighborhood ranking information based on data topology [31].

2.2. Current multi-hashing methods

Performances of hashing can be improved by either or both increasing the number of hash bits or/and the number of hash tables. The Complementary Hashing (CH) [32], the Dual Complementary Hashing (DCH) [33], the Boosting Iterative Quantization hashing with query-adaptive re-ranking (BIQH) [34], and the QsRank [35] are instances of multi-hashing methods. Both the CH [32] and the DCH [33] are boosting-based multi-hashing methods. The CH trains hash table to complement error made by the previous hash table while the DCH applies extra boosting during the training of hash bits. The DCH applies the SPLH to train individual hash table. Such that, in addition to complementing error made by the pervious hash bit in the training of the new hash bit in a hash table, the DCH also complements error made by the previous hash table during the training of a new hash table. The BIQH [34] constructs multiple ITQ hash tables by boosting and re-ranks retrieved images using a bit-level weight for each category in the image database. When a query image arrives, the query weight is computed by a weighted (portion of images in its category over all categories) average of category weights of top-N images of retrieved images. The major drawback of the BIQH is the requirement of fully labeled image database which may not be feasible for real-world large scale image retrieval problems. The QsRank [35] constructs undirected graphs using the relationship between the given query and an image in the database. The final retrieval result of the QsRank is constructed using a graph-based ranking method. The Multi-Graph Hashing [36] finds a weight for each graph and the final retrieval result is found by the combination of graphs and their weights.

In summary, current major multi-hashing methods (e.g. the CH, the DCH, and the BIQH) are boosting-based which may not be able to improve retrieval results by increasing the number of hash tables (after a small number). Boosting methods focus on learning of under-learned samples by new hash table and the number of under-learned samples reduces significantly after a small number of iterations. On the other hand, unsupervised hashing methods usually cannot achieve satisfying performance while supervised hashing methods requiring large scale image database to have all images being labeled may be impractical. Therefore, the bagging–boosting-based semi-supervised multi-hashing (BBSHR) method is proposed in this paper to relieve these. The BBSHR increases weights to under-learned samples instead of removing well-learned samples and uses semi-supervised hashing to fully utilize semantic information in partially labeled images.

3. The BBSHR

The Bagging–Boosting-based Semi-supervised Hashing with query-adaptive Re-ranking (BBSHR) consists of three major components: a hybrid semi-supervised multi-hashing to train multiple hash tables, semi-supervised category-specific weight generation, and a semi-supervised query-adaptive re-ranking to order the retrieved images for a given query. These three components will be proposed in Sections 3.1–3.3, respectively.

3.1. Hybrid semi-supervised multi-hashing for hashing tables construction

The proposed hybrid method uses a bagging approach to create multiple databases for training multiple hash tables (Section 3.1.1) and a boosting approach to train hash functions in each hash table (Section 3.1.2). In this way, we relieve the problem of severely reduced number of training samples of boosting-based multi-hashing methods after a small amount of training iterations.

3.1.1. Bagging-based semi-supervised multi-hashing

Bagging is widely used in machine learning for both classification and regression [37,38]. In general, the bagging method creates m training databases by randomly drawing pn images for m times from the unlabeled part of the original database X with n