Computer Vision and Image Understanding 000 (2016) 1–12

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Online supervised hashing

Fatih Cakir*, Sarah Adel Bargal, Stan Sclaroff

Department of Computer Science, Boston University, Boston, MA 02215, United States

A R T I C L E I N F O

Article history:
Received 15 January 2016
Revised 19 September 2016
Accepted 19 October 2016
Available online xxx

Keywords:
Hashing
Fast similarity search
Approximate nearest neighbors
Retrieval

A B S T R A C T

Fast nearest neighbor search is becoming more and more crucial given the advent of large-scale data in many computer vision applications. Hashing approaches provide both fast search mechanisms and compact index structures to address this critical need. In image retrieval problems where labeled training data is available, supervised hashing methods prevail over unsupervised methods. Most state-of-the-art supervised hashing approaches employ batch-learners. Unfortunately, batch-learning strategies may be inefficient when confronted with large datasets. Moreover, with batch-learners, it is unclear how to adapt the hash functions as the dataset continues to grow and new variations appear over time. To handle these issues, we propose OSH: an Online Supervised Hashing technique that is based on Error Correcting Output Codes. We consider a stochastic setting where the data arrives sequentially and our method learns and adapts its hashing functions in a discriminative manner. Our method makes no assumption about the number of possible class labels, and accommodates new classes as they are presented in the incoming data stream. In experiments with three image retrieval benchmarks, our method yields state-of-the-art retrieval performance as measured in Mean Average Precision, while also being orders-of-magnitude faster than competing batch methods for supervised hashing. Also, our method significantly outperforms recently introduced online hashing solutions.

1. Introduction

Given a query, finding similar points in a corpus is a central problem in many computer vision applications. The ever-growing size of available data collections and the increasing use of high-dimensional representations in describing data, have increased the computational complexity for performing similarity search, urging researchers to develop search strategies that can be used to explore such collections in an efficient and effective manner.

Various similarity search techniques have been proposed to address these challenges. Such techniques include tree-based construction algorithms (Arya et al., 1998; Jagadish et al., 2005), which partition the search space so that only a subset of data points is considered for a query. Another group of techniques employ dimensionality reduction methods (Roweis and Saul, 2000; Tenenbaum et al., 2000), which map the data to a lower-dimensional space while preserving the neighborhood structure. A speedup is achieved in distance computations with the more compact, lower-dimensional representations.

However, these approaches do not scale well with higher-dimensional data representations and larger datasets. One promising family of approaches is based on hashing, in which the data is mapped to binary vectors in Hamming space. The binary vector representations permit fast search mechanisms with a very small memory footprint. Example applications that utilize hashing include: image annotation (Wang et al., 2014b), visual tracking (Li et al., 2013), 3D reconstruction (Cheng et al., 2014), video segmentation (Liu et al., 2014), object detection (Dean et al., 2013) and multimedia retrieval (Gao et al., 2015; Song et al., 2015; 2013).

Hashing methods can be broadly categorized as data-independent and data-dependent techniques. Data-independent methods (Datar et al., 2004; Gionis et al., 1999; Kulis and Grauman, 2009) give guarantees on the approximation to particular metrics, without regard to the dataset that is to be indexed. However, for certain application settings, distances are defined only on the available data set; thus, data-dependent solutions (Gong and Lazebnik, 2011; Kulis and Darrell, 2009; Lin et al., 2013; Liu et al., 2012; Wang et al., 2012; Weiss et al., 2008) are formulated to learn the hashings directly from data.

Data-dependent methods generally outperform data-independent solutions in retrieval tasks primarily due to the training phase where desirable properties such as compactness and informativeness of the hash mapping are imposed. Consequently, the resulting binary codes better capture the data-distribution. However, this training phase usually involves solving a complex optimization problem in which the optimum is gener-

* Corresponding author.
E-mail addresses: fcakir@bu.edu (F. Cakir), sbargal@bu.edu (S.A. Bargal), sclaroff@bu.edu (S. Sclaroff).

http://dx.doi.org/10.1016/j.cviu.2016.10.009
1077-3142/

ally found via batch learning. This batch learning usually has time complexity that scales as a quadratic function of the number of items in the dataset. As a result, it is very costly to re-run the batch learning with each update of the corpus, in order to adapt the hash mapping for evolving data distributions. A static corpus is rarely observed in practice; on the contrary, expansions and diversifications of the data are very common. Data points associated to previously observed or unobserved classes may arrive, necessitating an update in the hash mapping to accommodate to this non-stationarity. In such cases, it would be extremely costly to repeatedly do batch learning from scratch.

Many hashing studies conform to two important properties initially stated by Weiss et al. (2008): (1) the domain of the hash mapping should cover the entire input space and (2) compact codes should be adequate enough in representing the data. Given the above discussion, we suggest a third important property: (3) a hash mapping must be amenable to the variation of a dataset. In this work, we propose an online supervised method for learning hash codes that satisfies all three properties. We specifically consider the problem of retrieving semantically similar neighbors where the semantics is induced from label information. This problem is central in many vision tasks including, but not limited to, label-based image retrieval and annotation (Carneiro et al., 2007; Guillaumin et al., 2009), semantic segmentation (Liu et al., 2011), image super resolution (Yue et al., 2013), etc. Supervised hashing methods have shown to outperform unsupervised methods in semantic retrieval mainly due to leveraging available label information.

Our formulation is based on Error Correcting Output Codes (ECOCs). ECOCs have their origins in coding theory and have been successfully used to solve many computer vision problems (Jiang and Tu, 2009; Kittler et al., 2001; Schapire, 1997; Zhao and Xing, 2013). The general theme is to use a distributed representation for the output space. These representations are carefully selected so they partition the output space into distant target regions. Errors made in the system (e.g., *channel* or *classifiers*) then can be recovered to a certain extent. In the hashing context, an ECOC formulation has several advantages. It allows one to be more specific regarding the *range* of the hash mapping. Prior work usually enforce properties to the hash mapping Φ through binary constraints resulting in complex (NP-hard) optimization problems. Instead, we directly construct the elements of the range (set) as desired and then do minimization. ECOCs also enables compensation for a number of hash function errors during retrieval when the range of Φ is carefully constructed. Finally, it provides a constant time hash-lookup complexity during retrieval.

We consider a stochastic setting in which data items sequentially arrive and the hash mapping is updated accordingly. The data items may be associated with previously unobserved labels; thus, we assume that the number of labels is not known *a priori*. In experimental evaluation, our proposed method yields accuracy that is at least comparable to (sometimes even better than) state-of-the-art batch solutions but is orders-of-magnitude faster in learning the hash mapping. Most importantly, our method is adaptable to data variations. This is critical for diversifying and expanding datasets (please observe Fig. 1). We also significantly outperform two competing recent online hashing methods (Cakir and Sclaroff, 2015; Huang Long-Kai and Wei-Shi, 2013).

In summary, our contributions are twofold:

1. We introduce an adaptive supervised hashing technique. It is orders-of-magnitude faster than state-of-the-art batch methods, while providing comparable or better accuracy. Also, compared to recently proposed online techniques, our method shows significant improvements.
2. Our learning formulation does not require any prior assumptions on the label space and is well-suited for expanding datasets that have new label inclusions. Our learning algorithm has linear time complexity with respect to number of items in the dataset. To the best of our knowledge, it is the first supervised hashing technique that allows the label space to grow.

The remainder of the paper is organized as follows. Section 2 briefly surveys related work. Section 3 gives the formulation of our methodology. Section 4 provides experiments followed by concluding remarks in Section 5.

2. Related work

In this section, we briefly provide a review of related hashing techniques.

2.1. Hashing

Many hashing methods have been introduced over the years. Notable earlier examples include Locality Sensitive Hashing methods (Datar et al., 2004; Gionis et al., 1999; Kulis and Grauman, 2009) where metric functions such as the Euclidean, Jaccard and Cosine distances are approximated. These methods usually have theoretical guarantees on the approximation quality and conform with sub-linear retrieval mechanisms. However, they are confined to certain metrics as they ignore the data distribution and/or related meta-data.

Contrary to earlier methods, recent approaches are data-dependent such that hash functions are directly learned from the data. These methods can be considered as binary embeddings that map the data into the Hamming space while preserving a specific neighborhood structure. Such a neighborhood is induced from the meta-data (e.g., labels) or is completely determined by the user (e.g., via similarity-dissimilarity indicators of data pairs). With the new binary representations, distance computations can be efficiently carried out allowing even a linear search to be done very efficiently for large-scale data. These data-dependent methods can be grouped as follows: rank preserving methods (Norouzi and Fleet, 2011; Shakhnarovich et al., 2003), similarity alignment techniques (Kulis and Darrell, 2009; Lin et al., 2014; Liu et al., 2012; Wang et al., 2012), quantization/PCA based methods (Gong and Lazebnik, 2011; He et al., 2013; Jegou et al., 2011), spectral and graph based solutions (Ge et al., 2014; Strecha et al., 2012; Weiss et al., 2008; Zhang et al., 2010), and very recently, deep-learning methods (Lai et al., 2015; Lin et al., 2015; Xia et al., 2014). We now review a few of the prominent methods. For a more comprehensive survey we refer readers to Wang et al. (2014a).

Among similarity alignment solutions, Minimal Loss Hashing (MLH) (Norouzi and Fleet, 2011) considers minimizing a hinge-type loss function motivated from structural SVMs. In Binary Reconstructive Embeddings (BRE), (Kulis and Darrell, 2009), a kernel-based solution is proposed where the goal is to construct hash functions by minimizing an empirical loss between the input and Hamming space distances via a coordinate descent type algorithm. Supervised Hashing with Kernels (SHK) (Liu et al., 2012) is similar to BRE such that a kernel-based solution is proposed; but, instead of preserving the equivalence of the input and Hamming space distances, the kernel function weights are learned by minimizing an objective function based on the binary code inner products.

Another notable line of work includes quantization/PCA based techniques. Among these, Semi-Supervised Hashing (Wang et al., 2012) learns the hash functions by maximizing the empirical accuracy on labeled data and also the entropy of the generated hash functions on any unlabeled data. This is shown to be very similar to doing a PCA analysis where the hash functions are the eigenvectors of the biased covariance matrix (biased due to the supervised

Please cite this article as: F. Cakir et al., Online supervised hashing, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2016.10.009