Neurocomputing 208 (2016) 59–65

Contents lists available at ScienceDirect

Neurocomputing

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

Binary code learning via optimal class representations

Xiang Zhou [a], Fumin Shen [a,b,*], Yang Yang [a], Guangwei Gao [c], Yuan Wang [d]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, PR China
[b] Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology), Nanjing, PR China
[c] Institute of Advanced Technology, Nanjing University of Posts and Telecommunications, Nanjing, PR China
[d] Department of Industrial and Systems Engineering, National University of Singapore, Singapore

ARTICLE INFO

Article history:
Received 30 October 2015
Received in revised form
20 December 2015
Accepted 22 December 2015
Available online 4 June 2016

Keywords:
Hashing
Binary code learning
Image retrieval

ABSTRACT

Hashing is an attracting technique for fast retrieval due to its low storage and computation costs. By hashing, each high-dimensional vector is mapped into a low-dimensional binary code vector and retrieval is performed in the Hamming space. Recently several hashing methods have been proposed, among which, supervised hashing methods have shown great performance by incorporating the supervision information. However, most previous supervised methods simply focused on the pairwise label information of data, and ignored the structure information and relationship within data. To tackle this problem, we propose to learn binary codes by explicitly taking into account class semantic relatedness. Specifically, a set of binary codes is computed according to the intrinsic class similarities in data and serves as the optimal class representations. We show that, by mapping images onto the optimal representation of their corresponding classes, our proposed method outperforms several other state-of-the-art supervised hashing methods in image retrieval on three large-scale datasets.
© 2016 Elsevier B.V. All rights reserved.

1. Introduction

With the explosive growth of images on the web, nearest neighbor search has attracted great attention in computer vision, machine learning, information retrieval and related area [5,16,8,37,18,17,19,48,38]. When the images are high dimensional, searching efficiently becomes challenging and crucial. Two mainstream retrieval approaches are *tree based* and *hashing based* methods. With tree based methods, searching is speeded up by exploiting spatial partitions of data space via various tree structures. Decision trees [27] and *kd*-trees [23] are two such methods. However, storage and time consumption grow exponentially with dimension growing, which lead to an inefficient search.

To search high-dimensional data efficiently, hashing becomes a promising approach. Hashing methods map the high-dimensional vector onto a low-dimensional binary code vector, and the mapped binary codes are used for efficient search. Besides search, binary code has also been widely applied in various vision applications [21,20,22]. Existing hashing techniques can be divided into two categories: *data-dependent* and *data-independent*. Locality Sensitive Hashing (LSH) [5] is one of the most popular *data-independent* methods. In LSH, the random hyperplane-based hash function involves a random projection sampled from a Gaussian distribution. In addition to Euclidean distance, Developed LSH methods employed several other distance measures such as *p*-norm distances [2], the Mahalanobis metric [12], and kernel versions [11,28]. The LSH family, however, needs long binary codes for achieving high search performances, which lead to a high storage consumption.

Instead of generating hash functions randomly, *data-dependent* hashing methods learn similarity-preserving binary code from training data. Various *data-dependent* methods have been proposed in the literature. Representative methods in this category can be divided into two parts: *supervised methods* and *unsupervised methods*. Unsupervised methods use the sole unlabelled data to generate binary codes. For example, PCA Hashing [47], ITQ [6], Isotropic hashing [9], Spectral Hashing (SH) [41] and Asymmetric Inner Product Binary Coding (AIBC) [29], are some widely used methods. These unsupervised methods, however, do not consider the supervision information. Therefore many supervised methods are proposed to handle this issue such as the supervised minimal loss hashing (MLH) [24], kernel-based supervised hashing (KSH) [15], supervised discrete hashing (SDH) [30], FastHash [13], graph cuts coding (GCC) [4] etc.

A few hashing methods propose to generate the hash functions in the kernel space as the extension of linear hashing methods, such as binary reconstructive embeddings (BRE) [10], KLSH [11]. Recently, it is shown that compact similarity-preserving hash

* Corresponding author at: School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, PR China.
E-mail addresses: Johnfly2000@gmail.com (X. Zhou), fumin.shen@gmail.com (F. Shen), dlyyang@gmail.com (Y. Yang), csggao@gmail.com (G. Gao), iseway@nus.edu.sg (Y. Wang).

http://dx.doi.org/10.1016/j.neucom.2015.12.129
0925-2312/© 2016 Elsevier B.V. All rights reserved.

codes can be obtained by considering the non-linear manifold structure. One of the most popular methods in this category is spectral hashing [41], which generates hash codes by solving the relaxed mathematics program that is similar to Laplacian eigenmaps [1]. As the extension of SH, anchor graph hashing (AGH) [16] use the anchor graph affinity, which makes training and the out-of-sample extension problem tractable for large-scale dataset. Inductive manifold hashing (IMH) [31,32] proposed a new framework for generating nonlinear hash functions. Other related methods include the multidimensional spectral hashing (MDSH) [40] and DGH [14].

In general, supervised methods outperform unsupervised methods due to the usage of supervision information of the training data. In SSH, a matrix S is defined incorporating the pairwise labeled information. In SDH, the label information is used for classifying binary codes. However, most existing supervised methods simply focus on the label information and pay no attention to the relationship between classes. We believe that the semantic relationship between classes gives more detailed and specific information than label information, and improve the retrieval performance.

In this work, we propose a new method to compute the binary codes for classes as the optimal representations, assuming that an optimal representation can be representative for its corresponding class and reflect the relationship with other classes. By considering the semantic similarity between classes, a matrix is constructed to depict the semantic similarity between classes. Thus, the optimal class representations are computed according to this matrix. Our contributions are as follows:

1. We propose a new supervised hashing method that each class is assigned to an optimal binary code as their class representation, considering that optimal class representations preserve semantic similarity between classes well.
2. We construct a semantic relatedness matrix to depict the semantic similarity, and then a set of binary codes is computed to preserve the similarity in the Hamming space. The binary codes of data are expected to be close with their corresponding optimal class representation. After solving a straightforward optimization problem, the binary codes and hash functions are learned efficiently.

2. Learning the optimal class binary representations

Suppose that we have n samples $X=\{x_i\}_{i=1}^{n}$. Our aim is to get a set of binary codes $B=\{b_i\}_{i=1}^{n}\in\{-1,1\}^{n\times L}$ to preserve their semantic similarities well. Here we want to get the optimal class representations which can capture the semantic similarities between classes, and then the set of binary codes B is learnt according to the corresponding optimal class representations.

For c classes, we compute a matrix $P=[p_1^T;p_2^T;\dots;p_c^T]\in\{1,-1\}^{c\times L}$, and every row p_i^T in P is the optimal class representation assigned to class i. In this and next sections, we will give the details about how to compute the optimal class representations and learn the binary codes of data according to the optimal class representations.

2.1. Class semantic relatedness

One important property of optimal class binary representations is that the class representation of semantically similar classes should be more close and dissimilar classes should be more far. In other words, the binary codes of two similar classes share more common bits, and two dissimilar classes share less common bits. In [49], the authors designed the semantic relatedness matrix to measure the similarity between classes. Thus, firstly we should compute the semantic relatedness matrix S to depict the similarity between classes. Semantic relatedness matrix S measures similarity between classes. Suppose there are two classes $\mathcal{X}_i=\{X_1^{(i)},\dots,X_{|\mathcal{X}_i|}^{(i)}\}$ and $\mathcal{X}_j=\{X_1^{(j)},\dots,X_{|\mathcal{X}_j|}^{(j)}\}$. The semantic similarity between $\mathcal{X}_i$ and $\mathcal{X}_j$ can be defined in many ways, such as *Hausdorff distance*, *match kernel* [7,25], *divergence between probability distributions* [26]. In this work, we use some match kernel in [7], and semantic similarity between $\mathcal{X}_i$ and $\mathcal{X}_j$ that can be expressed as follows:

$$S_{ij}=\frac{1}{|\mathcal{X}_i|}\frac{1}{|\mathcal{X}_j|}\sum_{p=1}^{|\mathcal{X}_i|}\sum_{q=1}^{|\mathcal{X}_j|}K(X_p^{(i)},X_q^{(j)}) \tag{1}$$

where $K(\cdot,\cdot)$ is a Mercer kernel and $|\cdot|$ is the cardinality of the class $\mathcal{X}_i$.

The semantic relatedness between $\mathcal{X}_i$ and $\mathcal{X}_j$ is the sum of $K(X_p^{(i)},X_q^{(j)})$, which could be computed inefficiently for large-scale dataset. In this work, we adopt a simple yet efficient linear kernel, and show that the semantic relatedness matrix can be computed in a very simple and compact way:

$$S_{ij}=\frac{1}{|\mathcal{X}_i|}\frac{1}{|\mathcal{X}_j|}\sum_{p=1}^{|\mathcal{X}_i|}\sum_{q=1}^{|\mathcal{X}_j|}K(X_p^{(i)},X_q^{(j)})=\frac{1}{|\mathcal{X}_i|}\frac{1}{|\mathcal{X}_j|}\sum_{p=1}^{|\mathcal{X}_i|}\sum_{q=1}^{|\mathcal{X}_j|}X_p^{(i)}T\cdot X_q^{(j)}=\left(\frac{1}{|\mathcal{X}_i|}\sum_{p=1}^{|\mathcal{X}_i|}X_p^{(i)}T\right)\cdot\left(\frac{1}{|\mathcal{X}_j|}\sum_{q=1}^{|\mathcal{X}_j|}X_q^{(j)}\right)=\overline{\mathcal{X}_i}^T\cdot\overline{\mathcal{X}_j} \tag{2}$$

Where $\overline{\mathcal{X}_i}$ and $\overline{\mathcal{X}_j}$ are the mean vectors of $\mathcal{X}_i$ and $\mathcal{X}_j$.

From problem (2), we can see that the similarity between $\mathcal{X}_i$ and $\mathcal{X}_j$ is the inner product of the mean vectors of two classes, which can be computed easily even for the large-scale dataset such as ImageNet.

S_{ij} depicts the similarity between class $\mathcal{X}_i$ and class $\mathcal{X}_j$. When S_{ij} is large, we want the inner product of P_i and P_j is large, otherwise small. Hence, the problem can be formulated as follows:

$$\max\sum_{i=1}^{c}\sum_{j=1}^{c}P_i^TP_jS_{ij}=\mathrm{tr}(PP^T\odot S)=\mathrm{tr}(P^TSP)$$
$$\text{s.t. } P\in\{1,-1\}^{c\times l} \tag{3}$$

Where $\mathrm{tr}(\cdot)$ is the trace norm, $\odot$ is the Hadamard product of two metrics, and c is the number of classes.

2.2. Orthogonal binary codes

Spectral Hashing [41] shows that requiring the bits to be uncorrelated leads to the maximal information. So we add the orthogonality constraints to class representations for enriching the information of class representations. Thus, the class representations can be computed as follows:

$$\min_P\left\|\frac{1}{l}P^TP-I\right\|^2 \tag{4}$$

where l is the length of optimal binary codes.

2.3. Final formulation

Combining (3) and (4) together, we have the following formulation for learning optimal binary codes:

$$\min_P\left\|\frac{1}{l}P^TP-I\right\|^2-\alpha\,\mathrm{tr}(P^TSP)$$
$$\text{s.t.} P\in\{-1,1\}^{c\times l} \tag{5}$$

The above problem can be reformulated as following by making

Download English Version:

https://daneshyari.com/en/article/4948603

Download Persian Version:

https://daneshyari.com/article/4948603

Daneshyari.com