J. Vis. Commun. Image R. 40 (2016) 847–851

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Unsupervised discriminative hashing ☆

Kun Zhan [a,*], Junpeng Guan [a], Yi Yang [a], Qun Wu [b]

[a] School of Information Science and Engineering, Lanzhou University, Lanzhou, China
[b] School of Art and Design, Zhejiang Sci-Tech University, Hangzhou, China

ARTICLE INFO

Article history:
Received 15 March 2016
Revised 10 July 2016
Accepted 20 August 2016
Available online 26 August 2016

Keywords:
Unsupervised discriminative hashing
Out-of-sample extrapolation
Manifold learning

ABSTRACT

Hashing is one of the popular solutions for approximate nearest neighbor search because of its low storage cost and fast retrieval speed, and many machine learning algorithms are adapted to learn effective hash function. As hash codes of the same cluster are similar to each other while the hash codes in different clusters are dissimilar, we propose an unsupervised discriminative hashing learning method (UDH) to improve discrimination among hash codes in different clusters. UDH shares a similar objective function with spectral hashing algorithm, and uses a modified graph Laplacian matrix to exploit local discriminant information. In addition, UDH is designed to enable efficient out-of-sample extension. Experiments on real world image datasets demonstrate the effectiveness of our novel approach for image retrieval.

1. Introduction

Hashing-based approximate nearest neighbor search has become popular due to its promising performance in terms of efficiency and accuracy [1,2]. The performance of nearest neighbors based algorithms can be significantly improved by exploiting a similarity measure and learning the similarity measure is closely related to the problem of feature learning [3–5].

A feasible way is to embed high-dimensional features into a low-dimensional Hamming space where similar items can be efficiently searched [6], which is usually performed by multiplying the feature by a projection matrix, subtracting a threshold and retaining the sign of the result. Locality sensitive hashing (LSH) algorithms are proposed for an approximate nearest neighbor search [7–9], but LSH is not stable and leads to bad results due to its randomized approximate nearest neighbor search and data-independent nature. The performance of date-dependent hash functions based on machine learning techniques is better than data-independent ones. Spectral hashing (SH) is a coding consistency hashing algorithm and requires small bits [10]. The assumption of uniformly distributed date does not hold in most cases resulting in that the performance of SH is deteriorated. He et al. extend SH by defining the hash function using kernels [11], Zhuang et al. extend SH from ordinary graph to hypergraph [12,13]. Sparse spectral hashing integrates sparse principal component analysis [14] and boosting similarity sensitive hashing into SH [15]. LSH relies on random projections and SH assumes features with uniformly distributed, which are problematic limitations. To avoid these limitations, there are many methods are proposed by using *kernel functions* to improved their performance [16–18].

Besides using manifold information as SH, we further consider the discriminative information into hash learning. Manifold learning is a suitable strategy to learn the embedding matrix from the manifold. Most manifold learning algorithms directly utilize Gaussian function to compute Laplacian matrix, which suffers from polytrope of bandwidth parameter. The discriminant information is not sufficiently exploited in aforementioned methods, so we construct a local clique comprising the data point and its neighboring data points in a nonlinear manifold by using local discriminant models and global integration (LDMGI) [19]. LDMGI is exploited by both manifold structure and local discriminant information simultaneously [19–23]. In our unsupervised discriminative hashing (UDH) algorithm, we use the LDMGI Laplacian matrix to learn hash codewords by using both manifold information and discriminant information, and the out-of-sample problem is addressed by the projection matrix which is computed during the hashing learning process.

In summary, the main contribution of this paper is twofold:

1. An unsupervised discriminative hashing algorithm is proposed.
2. We use an addition term as regularization to learn a model for out-of-sample data extrapolation.

The rest of this paper is organized as follows: In Section 2, we deduce our novel approach and give the solution of our approach and the algorithm to solve the regression framework. The

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: kzhan@lzu.edu.cn (K. Zhan).

http://dx.doi.org/10.1016/j.jvcir.2016.08.016
1047-3203/

experimental setting and analysis of results are showed in Section 3. The conclusion and discussion of future work are given in Section 4.

2. Unsupervised discriminative hashing

2.1. Preliminaries

Suppose that there are n training data points $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n]\in\mathbb{R}^{d\times n}$. $H=[\boldsymbol{h}_1,\boldsymbol{h}_2,\ldots,\boldsymbol{h}_n]^\top\in\mathbb{B}^{n\times m}$ denotes binary hash code of length m. $A\in\mathbb{R}^{n\times n}$ is the affinity matrix defined by $a_{ij}=\exp(-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/\sigma^2)$ and σ defines the standard deviation.

Spectral hashing (SH) seeks compact binary codes for a given data point where the similarity of data are preserved [10]. The objective function of SH is:

$$\begin{aligned}&\min \sum_{ij}A_{ij}\|\boldsymbol{h}_i-\boldsymbol{h}_j\|^2\\&\text{s.t. } \boldsymbol{h}_i\in\{-1,1\}^m,\sum_i\boldsymbol{h}_i=0,\\&\quad\frac{1}{n}\sum_i\boldsymbol{h}_i\boldsymbol{h}_i^\top=I\end{aligned}\tag{1}$$

where the constraint $\frac{1}{n}\sum_i\boldsymbol{h}_i\boldsymbol{h}_i^\top=I$ requires the bits to be uncorrelated.

By utilizing the spectral relaxation, (1) is rewritten by,

$$\begin{aligned}&\min \mathrm{Tr}(H^\top(D-A)H)\\&\text{s.t. } H_{ij}\in\{-1,1\},\\&\quad H^\top\mathbf{1}=0,H^\top H=I\end{aligned}\tag{2}$$

where $\mathrm{Tr}(\cdot)$ is trace operator, D is a diagonal matrix and its elements are column sums of A, $d_{ii}=\sum_j a_{ij}$. The codewords can be obtained by the m eigenvectors of $D-A$ with minimal eigenvalue.

2.2. Objective function

Many objective functions of manifold learning algorithms can be uniformly formulated by [20,24],

$$\begin{aligned}&\min_Y \mathrm{Tr}(Y^\top LY)\\&\text{s.t. } Y^\top Y=I\end{aligned}\tag{3}$$

where $Y=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_n]^\top\in\mathbb{R}^{n\times m}$ denotes the low dimensional embedding of X, $L=A-D$ is the graph Laplacian matrix [25].

The Laplacian matrix plays a very important role in manifold learning algorithms. Different from the affinity matrix in existing manifold learning algorithms is usually pre-computed among nearby data pairs by a fixed function, *e.g.*, the RBF kernel, we construct the Laplacian matrix taking account to both the discriminant information and the manifold structure of data [19]. To globally integrate the local discriminant models from all the cliques, the Laplacian matrix is constructed by,

$$L=\sum_{i=1}^n S_iL_iS_i^\top=[S_1,S_2,\ldots,S_n]\begin{bmatrix}L_1&&&\\&L_2&&\\&&\ddots&\\&&&L_n\end{bmatrix}[S_1,S_2,\ldots,S_n]^\top\tag{4}$$

where L_i is a positive semi-definite matrix $L_i=H_k(H_k^\top X_i^\top X_iH_k+\lambda)^{-1}H_k, X_i=[\boldsymbol{x}_0,\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{k-1}]$ is made up of $\boldsymbol{x}_i$ and its $k-1$ nearest neighbors is the local data matrix comprising all the data points in $\mathcal{N}_k(x_i)$. $S_i\in\mathbb{B}^{n\times k}$ is the selection matrix with its element $(S_i)_{pq}=1$, if $p=F_i\{q\}$; $(S_i)_{pq}=0$, otherwise. $F_i=\{i_0,i_1,\ldots,i_k\}$ denotes the index set of the samples in $\mathcal{N}_k(x_i)$. H_k is the centering matrix,

$$H_k=I-\frac{1}{k}\mathbf{1}\mathbf{1}^\top\tag{5}$$

where $I\in\mathbb{R}^{k\times k}$ is an identity matrix and $\mathbf{1}$ is the column-vector consisting of k ones.

In order to enable out-of-sample extension, we assume

$$Y=X^\top W.\tag{6}$$

By $\boldsymbol{y}_{\text{new}}=\boldsymbol{x}_{\text{new}}^\top W$, we can predict the output $\boldsymbol{y}_{\text{new}}$ if a new test data point $\boldsymbol{x}_{\text{new}}$ is input.

Y is defined by (6) as a linear regression model, so we given the following regression function to learn W,

$$\min_W \|X^\top W-Y\|_{\mathrm{F}}^2+\beta\|W\|_{\mathrm{F}}^2.\tag{7}$$

We incorporate (7) as an additional term of (3), then we obtain,

$$\begin{aligned}&\min_{Y,W}\mathrm{Tr}(Y^\top LY)+\alpha\|X^\top W-Y\|_{\mathrm{F}}^2+\beta\|W\|_{\mathrm{F}}^2\\&\text{s.t. } Y^\top Y=I\end{aligned}\tag{8}$$

where the α and β are two regularization parameters.

When α tends to zero, (8) becomes (3) which can learn the nonlinear Y. When α tends to infinity, $\|X^\top W-Y\|_{\mathrm{F}}^2$ equal to zero due to the minimization. So $X^\top W-Y=0$ and (8) becomes (7).

After learning the embedding matrix Y, the hash code H can be obtained by:

$$H=\mathrm{sign}(Y),\tag{9}$$

where $\mathrm{sign}(\cdot)$ is the sign function which makes the value binary.

2.3. Algorithm derivation

The Lagrangian function of (8) is,

$$\mathcal{L}(W,Y,v)=\mathrm{Tr}(Y^\top LY)+\alpha\|X^\top W-Y\|_{\mathrm{F}}^2+\beta\|W\|_{\mathrm{F}}^2+v\mathrm{Tr}(I-Y^\top Y)\tag{10}$$

where v is the Lagrangian multiplier.

The optimum Y and W can be obtained by calculating the first order derivative (8) with respect to Y and W, respectively. By setting the derivative to zero, we have,

$$\frac{\partial\mathcal{L}}{\partial W}=2\alpha XX^\top W-2\alpha XY-2\beta W=0.\tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial Y}=2LY-2\alpha X^\top W+2\alpha Y-2vY=0.\tag{12}$$

From (11), we obtain,

$$W=\left(XX^\top-\frac{\beta}{\alpha}I\right)^{-1}XY=MY,\tag{13}$$

where M is denoted by,

$$M=\left(XX^\top-\frac{\beta}{\alpha}I\right)^{-1}X.\tag{14}$$

From (12), we obtain,

$$LY-\alpha X^\top W+\alpha Y=vY.\tag{15}$$

Substituting (13) into (15), we obtain

$$(L-\alpha X^\top M+\alpha I)Y=vY.\tag{16}$$

The optimal solution Y of (8) is formed by the m eigenvectors of the term $L-\alpha X^\top M+\alpha I$ corresponding to the m smallest eigenvalues.