Image Retrieval Algorithm Based on Minimal Loss Hashing

Biao Wang¹, Ying Qian²

¹Graphic image and multimedia lab, Chongqing University of Posts and Telecommunication, Chongqing, China

² Graphic image and multimedia lab, Chongqing University of Posts and Telecommunication, Chongqing, China

Abstract. In order to solve the inefficiency and time-consuming of traditional image retrieval algorithms, an image retrieval algorithm based on minimal loss hashing is proposed. Firstly, the original high dimensional data is reduced by principal component analysis and Laplacian Eigenmaps. Secondly, minimize dimensionality reduction and quantization coding loss function, then we could obtain the hash function by iterative optimization parameters. Finally, the original data matrix is converted into a hash coding matrix, and the sample similarity is obtained by calculating the Hamming distance between samples. The experimental results on four public datasets show that the proposed method improves the retrieval performance.

1 INTRODUCTION

With the development of big data, content-based image retrieval technology becomes more and more important when dealing with large-scale image data. Traditional nearest neighbour search based on linear scanning usually becomes computationally prohibitive when the dataset is large. In order to reduce the space complexity and time complexity of retrieval, a hash-based nearest neighbour retrieval method is proposed.

The hash-based image retrieval method can encode the original features into compact binary hash codes, so that the hash-based image retrieval method can greatly reduce the memory consumption. When calculating the Hamming distance, an XOR operation of computer internal calculator can be used to make calculation of the Hamming distance within the microseconds, reducing the time required for single query.

In 1999, Locality-Sensitive Hashing (LSH) [1] was proposed by Piotr Indyk et al. It was the earliest image hashing algorithm. And it has better time efficiency and maintains good performance in high-dimensional space. However, a random mapping method is used when the construction of hash function in the Locality-Sensitive Hashing algorithm, and it is independent of the data set, so that the effect is not stable in practical applications.

In order to overcome the shortage of Locality-Sensitive Hashing, data-dependent hashing method is proposed. The hash function is generated from the data and solves the bottleneck problem of Locality-Sensitive Hashing. The existing data-dependent hash methods are mainly divided into supervised hash and unsupervised hash.

supervised hashing method utilize the extra information such as class labels or tags of the data to generate hash function. In BRE [2], the hashing functions are learned based on explicitly minimizing the

reconstruction error between the metric space and Hamming space. Semantic Hashing [3], which is the first work using a multi-layer Deep Neural Network for hashing, learns binary codes stacked with Restricted Boltzmann Machine to retain the semantic similarity structure of data. The hashing methods based on DNN develop fast with the growth of deep learning community, and various deep hashing methods have been proposed such as CNNH [4], DSH [5] and each other. The supervised hashing method is limited by its a priori information access problems, and unsupervised hashing method has no such fetters.

The unsupervised hash method uses the untagged training data to generate a binary code and maintains similar geometry by exploring the structural properties of the training data set. Weiss proposed spectral hash (SH) [6] method. It encodes the image feature vector which can be seen as a graph segmentation problem. By analysing the Laplacian eigenvalues and eigenvectors of similar graphs, a relaxation solution can be provided to the graph segmentation problem. However, this method assumes that the data is uniformly distributed in hyper-rectangles, the limits are too strict and difficult to achieve in practice. In order to solve the shortcomings of SH, a self-learning hash (STH) [7] is proposed, which uses the binarized Laplacian Eigenmaps to learn the hash function and makes SH suitable for any data distribution. Another effective method of unsupervised hashing, also known as iterative quantization (ITQ) [8], has been proposed to minimize the quantization error that maps zero-centre data through the optimal rotation matrix to the vertices of a binary hypercube. By the principle of ITQ, Yu et al. Propose a circular binary embedded hash (CBE) [9], which projects the data to a binary code with a circular matrix, which can be accelerated using Fast Fourier Transforms. SELVE encodes sparse embedding features by learning dictionaries and binarizes the

* Corresponding author: ¹ Biao Wang: 1055403145@qq.com

² Ying Qian: qianying@cqupt.edu.cn

encoding coefficients into hash codes. SpH [10] is different from the hyperplane-based LSH hash method, which defines a hypersphere in the original feature space, and performs binary coding on both sides of the ball. In addition, there are unsupervised hashing methods such as MCR [11], THH [12], GHS [13] and so on.

Most current hashing methods only use principal component analysis to reduce the dimension to protect the overall structure of data or preserve the local structure of data through Laplacian Eigenmaps. They failed to fully exploit the image features of the original data and did not take into account the loss of the dimensionality reduction when considering the quantization loss, which all hindered the improvement of image retrieval performance. In order to solve the above problem, this paper proposes a method of image retrieval with minimum loss hash, which combines feature analysis with Laplacian Eigenmaps to reduce dimensionality loss and reduce information loss by considering quantization loss and dimensionality loss.

2 CONSTRUCTION OF MINIMAL LOSS HASHING FUNCTION

The algorithm in this paper mainly consists of two parts. In the first step, the reduction dimensionality matrix T is obtained by combining the dimension reduction schemes of ITQ [8] and SH [6]. In the second step, the loss function of feature reduction and quantization coding is obtained, and the hash function is obtained through minimization. The first step can be the original high-dimensional data into low-dimensional data, the second step convert low-dimensional data into binary codes.

2.1 Notations and Definition

We have a set of n data points, $\{x_1, x_2, \dots, x_n\}$, $x_i \in \mathbb{R}^d$, that form the rows of the data matrix $X \in \mathbb{R}^{n \times d}$. We assume that the points are zero-centered, i.e., $\sum_{i=1}^n x_i = 0$. $\text{sgn}(v)$ Defined as a binarization function and written as follows:

$$\text{sgn}(v) = \begin{cases} 1, & \text{if } v \geq 0, \\ -1, & \text{otherwise} \end{cases} \quad (1)$$

Our goal is to learn a binary code matrix $B = \{-1, 1\}^{n \times k}$, where k denotes the code length. The original dataset $X \in \mathbb{R}^{n \times d}$ in Euclidean space is encoded to binary codes as $B \in \mathbb{R}^{n \times k}$ in Hamming space, while preserving semantic structure of data points.

2.2 Dimensionality Reduction

PCA (Principal Component Analysis) method is a common data analysis method, which converts the original data into a set of linearly independent representations of each dimension by linear transformation. By selecting the eigenvectors corresponding to the k largest eigenvalues of the covariance matrix Re-combined into a low-dimensional

matrix, to achieve the purpose of dimension reduction. The algorithm refers to the objective function [8] of reducing dimension:

$$\tilde{I}(T) = \frac{1}{n} \sum_k t_k^T X^T X t_k = \frac{1}{n} \text{tr}(T^T X^T X T) \quad \text{s.t. } T T^T = I \quad (2)$$

The PCA method is introduced into this algorithm model. Maximize (2) to get a large amount of information. we get the following objective function:

$$\max \text{tr}(T^T X^T X T) \quad \text{s.t. } T T^T = I \quad (3)$$

For a code of k bits, we obtain reduced dimension matrix $T \in \mathbb{R}^{d \times k}$ by taking the top k eigenvectors of the data covariance matrix $X^T X$.

LE (Laplacian Eigenmaps) is another way to reduce the dimension, it is from a local perspective to build the relationship between the data. If the two data points are similar, after dimensionality reduction they should be as close as possible in the target subspace. First of all, a similar structure matrix $S \in \mathbb{R}^{n \times n}$ of the original data is constructed to achieve the purpose of preserving the local similar structure of the data. Then define a diagonal matrix $D \in \mathbb{R}^{n \times n}$, The matrix after dimension reduction $V \in \mathbb{R}^{n \times k}$, The diagonal elements in the D matrix are equal to the sum of each row elements of the matrix S , i.e., $D_{i,i} = \sum_{j=1}^n S_{i,j}$. The Laplace matrix $L = D - S$ contains the local similarity structure of the original data. The objective function is as follows:

$$\min \text{tr}(V^T L V) \quad \text{s.t. } V^T D V = I \quad (4)$$

$\text{s.t. } V^T D V = I$ can guarantee optimization problems have solutions, And the data points after the mapping will not be "compressed" to a less than m -dimensional subspace, we obtain an easy problem whose solutions are simply the k eigenvectors of L with minimal eigenvalue.

For Low-dimensional matrix $V = XT$, (4) can be expressed as:

$$\min \text{tr}(T^T X^T L X T) \quad \text{s.t. } T^T T = I \quad (5)$$

Combined with (3) and (5), we can get the objective function:

$$\begin{aligned} & \min \text{tr}(T^T X^T L X T) - \lambda \text{tr}(T^T X^T X T) \\ & = \min \text{tr}(T^T X^T (L - \lambda I_n) X T) \quad \text{s.t. } T^T T = I \end{aligned} \quad (6)$$

The PCA of (3) can preserve the overall similarity structure of the original data, and the (5) can preserve the local similarity of the data. (6) preserves the local structure of the data as much as possible while preserving the overall structure of the data. This parameter λ is a tuning parameter used to balance the weights between two similar structures, and its value will be given later. Finally, the dimensionality reduction matrix T is obtained by solving the eigenvectors corresponding to the k smallest non-zero eigenvalues.

2.3 Loss function

This section mainly builds the loss function from the two aspects of reduction loss and binary quantization loss. For a given dataset X , we can approximate it with two matrices: $X \approx VT^T$. The loss of dimensionality reduction

can be expressed as the square of the Euclidean distance between the original matrix and the two matrices:

$$L_1 = \min \|X - VT^T\|_F^2 + \gamma (\|V\|_F^2 + \|T\|_F^2) \quad (7)$$

In order to improve the over-fitting phenomenon, add the regular term in the above formula. It can improve the generalization ability of the model. γ is the regular term balance coefficient.

After the original data is mapped into the low-dimensional space through dimension reduction, the process of quantization coding can be seen as the process of assigning each data point $v \in R^k$ to the vertices of the hypercube. Each vertex $\text{sgn}(v)$ of the hypercube is a binary vector $\{1, -1\}^k$. k is the length of the binary code, so the binary code for each data corresponds to a vertex of the hypercube. Quantitative loss can be expressed as $\|\text{sgn}(v) - v\|^2$. Minimizing the quantization loss can make the generated binary code retain more of the original neighbourhood of the data structure. In order to avoid the data after the projection in a direction of variance is too large, causing the corresponding code bits larger weight and other bits less weight, resulting in reduced search performance. Orthogonal matrix R is introduced into dimensionally-reduced data V to rotate it, and the weight of each hashing code is equalized to reduce the quantization loss. Thus, we can get the objective function:

$$L_2 = \min \|B - VR\|_F^2 \quad (8)$$

From the above loss function, L_1 and L_2 represent the dimensionality loss and the quantization loss, respectively. But the two losses are independent of each other. To reduce both of these losses in general, Lagrange multiplier methods are used to connect the two loss functions with the following objective function:

$$L = L_1 + \varepsilon L_2$$

$$= \min \|X - VT^T\|_F^2 + \gamma (\|V\|_F^2 + \|T\|_F^2) + \varepsilon \|B - VR\|_F^2 \quad (9)$$

The undetermined coefficient ε is Lagrange multiplier, which is used to balance the above two loss functions.

2.4 Function optimization

Expanding the objective function (9):

$$L = \text{tr} \left[(X - VT^T)(X - VT^T)^T \right]$$

$$+ \gamma (\|V\|_F^2 + \|T\|_F^2) + \varepsilon \text{tr} \left[(B - VR)(B - VR)^T \right]$$

$$= \text{tr}(XX^T) - 2\text{tr}(VT^T X^T) + \|VT^T\|_F^2 + \varepsilon \text{tr}(BB^T)$$

$$- 2\varepsilon \text{tr}(VRB^T) + \varepsilon \|V\|_F^2 + \gamma (\|V\|_F^2 + \|T\|_F^2) \quad (10)$$

In order to make the loss function smaller, a better hash function is obtained. L can find the partial derivatives of V , T , R and B respectively and find their corresponding values at the poles.

Follow the steps below to iterate until convergence. Initialization, random matrix $V \in R^{n \times k}$, $R \in R^{k \times k}$ orthogonal random matrix, $T \in R^{d \times k}$ is the reduced dimension matrix: 1. Fix V , T , R ,

$$\dot{L} = -2\varepsilon \text{tr}(BR^T V^T) + c, \quad c \text{ is a constant} \quad (11)$$

Minimization (11) is maximization $\text{tr}(BR^T V^T) = \sum_{i=1}^n \sum_{j=1}^k B_{ij} \tilde{V}_{ij}$. we need to

$B_{ij} = \begin{cases} 1 & \tilde{V}_{ij} \geq 0, \\ -1 & \text{otherwise} \end{cases}$. So B can be expressed as:

$$B = \text{sgn}(VR) \quad (12)$$

2. Fix T , R and B , take the derivative of objective function (10) with respect to V :

$$\frac{\partial L}{\partial V} = \frac{\partial [-2\text{tr}(VT^T X^T) + \|VT^T\|_F^2 - 2\varepsilon \text{tr}(VRB^T) + (\varepsilon + \gamma)\|V\|_F^2]}{\partial V}$$

$$= -2XT + 2VT^T T - 2\varepsilon BR^T + 2(\varepsilon + \gamma)V$$

$$= 2\{V(T^T T + (\varepsilon + \gamma)I) - (XT + \varepsilon BR^T)\}$$

namely $\frac{\partial L}{\partial V} = 0$, then V can be obtained:

$$V = (XT + \varepsilon BR^T)(T^T T + (\varepsilon + \gamma)I)^{-1} \quad (13)$$

3. Fix V , R and B , take the derivative of objective function (10) with respect to T :

$$\frac{\partial L}{\partial T} = \frac{\partial [-2\text{tr}(VT^T X^T) + \|VT^T\|_F^2 + \gamma\|T\|_F^2]}{\partial T}$$

$$= -2X^T V + 2TV^T V + 2\gamma T$$

$$= 2[T(VV + \gamma I) - X^T V]$$

Namely $\frac{\partial L}{\partial T} = 0$, then T can be obtained :

$$T = X^T V (V^T V + \gamma I)^{-1} \quad (14)$$

4. Fix V , T and B ,

$$\dot{L} = -2\varepsilon \text{tr}(BR^T V^T) + c, \quad c \text{ is a constant} \quad (15)$$

Minimize the loss function, which is equivalent to maximize $\text{tr}(BR^T V^T)$, use singular value decomposition to solve it. Decomposing $B^T V$ into the form of $S_1 \Omega S_2^T$ using singular values, then:

$$R = S_2 S_1^T \quad (16)$$

3 RESULTS AND DISCUSSION

3.1 Experimental datasets and evaluation criteria

The experiments in this paper are run under the Win10 system MATLAB2016a software, the system memory is 12GB, CPU 2.30GHz. The experimental datasets are described below:

CIFAR-10 dataset includes 60,000 colour images and divides the images into 10 categories, each of which contains 6,000 images. Each image is represented by a 512-dimensional eigenvector. In this experiment, 59000 images were chosen as train dataset, and the remaining 1000 images were used as the test dataset.

Caltech-256 dataset contains 29,780 images, the dataset is divided into 256 classes, and each class contains at least 80 images. In this experiment, we use the CNN feature of 1024 dimensions to represent, select 1000 as the test data set and the rest as the training set.

NUS-WIDE dataset contains 269,648 images, including six low-level features such as colour histogram, colour-related graph, edge-oriented histogram, wavelet texture, and patch colour moments. In this paper, we choose the block colour moment feature of this data set

with a dimension of 225, because the length of the binary code is generally not higher than the dimension of the feature, and the maximum length of the code is 128 in this paper.

MNIST Handwritten Digital Image Library dataset contains 70,000 images, each 28 * 28 in size, containing a total of 10 handwritten character images numbered from 0 to 9. We generally use the pixel gray value of the image directly as the image feature, and the number of dimensions is 784.

In order to measure the performance of the algorithm, we use Hamming sorting method. For images that need to be queried, we sort the images in the database by the Hamming distance from the query image. The quantitative evaluation of the retrieval performance is measured with four popular metrics: precision-recall curve, precision-the number of returned samples, recall-the number of returned samples and mean average precision (MAP). The definitions of precision, recall and MAP are as follows:

$$\text{precision} = \frac{\text{Number of retrieved relevant images}}{\text{Total number of all retrieved images}} \quad (17)$$

$$\text{recall} = \frac{\text{Number of retrieved relevant images}}{\text{Number of all relevant images}} \quad (18)$$

$$\text{MAP} = \frac{\sum_{q=1}^N \int_0^1 p(r) dr}{N} \approx \frac{\sum_{q=1}^N \sum_{k=1}^m P(k) \Delta r(k)}{N} \quad (19)$$

The MAP calculates the area between the PR curve and the axis and can be used to measure the overall performance of the algorithm.

3.2 Parameter Settings

In the parameter settings, MAP as the first performance measure to select the parameters. On the one hand, the choice of the balance parameter λ used to balance PCA and LE dimensionality reduction in Section 2.2. From Fig.1 (a), we can see the varies of λ from 0.00001 to 10 and the MAP varies with it. As we can be seen from the figure, the value of MAP takes a significant decrease when λ is taken as 0.001, so λ is taken as 0.001.

Next, we choose the loss function Lagrange multiplier ϵ and the regular term balance coefficient γ in Section 2.3. In Figure 1 (b), we can see that γ and ϵ varies from 0.0001 to 10, the contour map of MAP, so $(\gamma, \epsilon) = (0.01, 0.001)$.

Fig. 1. CIFAR-10 dataset: (a) MAP varies with respect to λ . (b) MAP contour plots varies with respect to ϵ and γ .

3.3 Results

1) Precision-Recall Curves: Figures 2 and 3 are PR curves for 32 bits, 64 bits and 128 bits of the algorithm on the CIFAR10 and CALTE256 datasets. It can be seen from the figure that the Precision and recall of all the methods increase with the increase of the code length, while the fixed code length Precision decreases with the increase of the recall rate, that is, the two are inversely proportional. From the figure we can see that our method is generally above the rest of the methods, it shows that the use of minimum hash method in these length encoding, we will get better results when we query images.

2) precision curves: This curve represents the curve of the corresponding precision after returning different numbers of images. The higher the precision curve, the higher the efficiency of the method. Figures 4 and 6 show the precision curves with

Fig.2. PR curves on CIFAR10.

Fig.3. PR curves on CALTE256.

different code lengths on the CIFAR10 and CALTE256 datasets, respectively.

Fig.4. Precision curves on CIFAR10.

Fig.5. Recall curves on CIFAR10

3) recall curves : The recall rate reflects the search engine's ability to return the correct image. Similar to the accuracy curve, the recall curve shows the corresponding recall rates across different numbers of returned images. Figure 5 and Figure 7 show the recall curves for different code lengths on the CIFAR10 and CALTE256 datasets, returning images from 0 to 1000. In contrast to the other methods, the recall rate of our method is still acceptable.

Fig.6. Precision curves on CALTECH256.

Fig.7. Recall curves on CALTECH256

4) MAP curve: MAP refers to the average precision of all the training samples retrieved. MAP is one of the most important indexes to evaluate the performance of image retrieval. The higher MAP value indicates the better retrieval performance. Figure 8 shows the different performance of different hash image retrieval methods for encoding MAP curves from 16 bits to 128 bits on four datasets CALTECH256, CIFAR10, NUS-WIDE and MNIST. In addition, it can be seen that the results of all the methods under the CIFAR-10 dataset have obviously decreased. This is because the images under the CIFAR-10 dataset are complicated by themselves. However, in the same test environment, Other methods have advantages.

Fig.8. MAP curves on CALTECH256, CIFAR10, NUS-WIDE and MNIST

In this paper, the minimum loss hash algorithm is proposed. The learned hash function performs better because it relies on the data first, and secondly preserves the global and local similarities of the data, while also taking into account the loss of information both in dimensionality reduction and binary coding. The experiment in this paper also clearly proved this point.

4 CONCLUSIONS

In this paper, we try our best to preserve the similar structure of data and reduce the loss of information and get the least loss hash algorithm. Firstly, this method combines both PCA and LE methods to obtain the

optimal dimensionality reduction matrix, and then minimizes the loss function. Finally, the optimized parameters are obtained through iteration, and the final hash function is obtained. Because the algorithm retains the local and global similar structures of the original data at the same time, the deficiencies of the single hash algorithm in preserving the data structure can be improved. At the same time, the loss of information can be reduced through the loss function in the entire quantization coding process. Experiments show that the minimal loss hash algorithm proposed in this paper has higher retrieval efficiency than the current mainstream hash algorithm.

References

1. Indyk P, Motwani R, *In Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC '98)*.604-613(1998)
2. Kulis B, Darrell T. *In Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS'09)*, Lafferty, C. K. I. Williams, and A. Culotta (Eds.). Curran Associates Inc.,1042-1050(2009)
3. Salakhutdinov R, Hinton G. *Int. J. Approx. Reasoning*, **50**, 969-978(2009)
4. Li W J, Wang S, Kang W C. *EprintarXiv:1511.03855.1711-1717*(2015)
5. Lin K, Yang H F, Hsiao J H. *Computer Vision and Pattern Recognition Workshops IEEE*,27-35(2015)
6. Weiss Y, Torralba A, Fergus R. *Conference on Neural Information Processing Systems, Vancouver*, 1753-1760(2008)
7. Zhang D, Wang J, Cai D. *In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR'10)*. ACM,18-25(2010)
8. Gong Y, Lazebnik S, Gordo A. *IEEE Transactions on Pattern Analysis & Machine Intelligence*,**35**,2916-2929(2013)
9. Yu F X, Kumar S, Gong Y, et al. *Proceedings of the 31st International Conference on Machine Learning*, PMLR **32**,946-954(2014)
10. Heo J P, Lee Y, He J, et al. *2012 IEEE Conference on Computer Vision and Pattern Recognition*.2957-2964(2012)
11. Guo Y, Ding G, Liu L. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*.1344-1354(2017)
12. Zhu L, Shen J, Xie L. *In Proceedings of the 23rd ACM international conference on Multimedia (MM '15)*. ACM,843-846(2015).
13. Tian D, Tao D. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*.79-89(2016).