Simhash for large scale image retrieval

Qin-Zhen Guo, Zhi Zeng, Shuwu Zhang, Xiao Feng and Hu Guan

95 Zhongguancun East Road, 100190, BEIJING, CHINA

{ qinzhen.guo, zhi.zeng, shuwu.zhang, xiao.feng, hu.guan }@ia.ac.cn

Keywords: Hashing, bag-of-visual-words, simhash, image retrieval.

Abstract.
Due to its fast query speed and reduced storage cost, hashing, which tries to learn binary code representation for data with the expectation of preserving the neighborhood structure in the original data space, has been widely used in a large variety of applications like image retrieval. For most existing image retrieval methods with hashing, there are two main steps: describe images with feature vectors, and then use hashing methods to encode the feature vectors. In this paper, we make two research contributions. First, we creatively propose to use simhash which can be intrinsically combined with the popular image representation method, Bag-of-visual-words (BoW) for image retrieval. Second, we novelly incorporate "locality-sensitive" hashing into simhash to take the correlation of the visual words of BoW into consideration to make similar visual words have similar fingerprint. Extensive experiments have verified the superiority of our method over some state-of-the-art methods for image retrieval task.

Introduction

With the rapid development of the Internet, the number of images on the web is growing explosively. It is getting more important to retrieval images that web users desire. Nowadays, content-based image retrieval (CBIR) is attracting more and more attention. Generally a CBIR system consists of two key components: an effective image representation and an efficient retrieval strategy.

The primal image representations for CBIR are global low-level features like color and texture. But global features do not capture local image information and usually they have poor discriminability. Bag-of-visual-words (BoW) [1], where local image descriptors (e.g., SIFT [2]) are extracted and quantized based on a set of visual words, is a popular image representation method and has better discriminability. For retrieval strategy, a naive solution to finding nearest neighbors is to search over all the database samples and sorting them according to their similarity to the query. This becomes prohibitively expensive when the size of database is large, especially when the dimensionality of image representation is high. However for image retrieval task, it is sufficient to find approximate nearest neighbors (ANN). In recent years, many ANN search techniques have been researched including tree-based methods and hashing-based methods. Since the tree-based methods can be degenerated to exhaustive search for high dimensions, hashing-based ANN search techniques [3, 4, 5, 6, 7, 8] which aim at mapping the data into binary codes in Hamming space where similarity can be measured by Hamming distance have attracted increasing attentions.

The pioneering work LSH [3] employs simple random hash functions to project the data and uses zero to binarize the projected data. Spectral hashing (SH) [4] calculates the bits by thresholding a subset of eigenvectors of the Laplacian of the similarity graph. In Hamming Embedding (HE) [5], Jégou et al. randomly draws a matrix of Gaussian values and apply QR factorization to it. Then use the first rows of the orthogonal matrix obtained by this decomposition to project the data followed by using median value to binarize every dimension. Similarly, in order to computes b-bit hash codes, PCAH [6] projects data to the b principal components, and then use average value to binarize the coefficients. A new work Isotropic Hashing (IsoH) [7] notices that different dimensions of the data after projection such as PCA have different dispersion and it is unreasonable to use the same number

of bits to encode different projected dimensions. In IsoH, the PCA-projected data are re-projected by a trained orthogonal matrix to guarantee the same dispersion of the different dimensions.

In this paper, we propose to use simhash for large scale image retrieval. Based on the naive simhash where the visual words (words, for simplicity) are encoded in cryptographic hashing method where the correlation of the words are not considered, we present a novel simhash method in which the words are encoded by "locality-sensitive" hashing.

Learning Simhash Codes

Overview of Simhash. Simhash is initially used by Google to detect near-duplicate documents [9]. It is a "locality-sensitive" hashing method. That is documents that have similar content will have similar simhash codes (codes that have a small Hamming distance) while disparate documents have markedly different simhash codes (codes that have a large Hamming distance).

The procedures of simhash for near-duplicate documents detection are as follows: Given a collection, $\boldsymbol{D}$, of terms (or words) t_i extracted from a documents and their corresponding weight w_i. Assuming that the desired total code length is b and $\boldsymbol{B}$ is a b-dimensional zero vector, for a document, we use the cryptographic hashing method (different documents have different hash values or hash codes, and even if they have similar content, the hash codes can be widely different) like MD5 or SHA-1 to encode every word t_i to b-bit fingerprint $\boldsymbol{s_i} \in \{-1,1\}^b$. If the j-th bit of $\boldsymbol{s_i}$ is 1, the j-th dimension of $\boldsymbol{B}$ is incremented by the weight of the corresponding word. Otherwise, the j-th dimension of $\boldsymbol{s_i}$ is decremented by the weight. After processing all the words, $\boldsymbol{B}$ will be binarized by zero. Then the j-th bit of simhash code will be

$$B_j = sign(\sum_{i=1}^{|D|} s_{ij} w_i), \tag{1}$$

$|D|$ is the cardinality of $\boldsymbol{D}$. s_{ij} is the j-th bit of $\boldsymbol{s_i}$ which is the fingerprint of word t_i.

Simhash for Image Retrieval. In order to use the framework of simhash for image retrieval, first of all, we should represent images by the collection of features. The most intuitive method is "Bag-of-visual-words" (BoW) where images are described by a set of visual words and in our approach, we represent images by BoW. For the weight of the words $\boldsymbol{t_i}$ (a vector, different from t_i in documents), we adopt the widely used *tf-idf* weight. The weight of word $\boldsymbol{t_i}$ in image p, $w_{i,p}$ is calculated by Eq. 2,

$$w_{i,p} = tf_{i,p} \cdot idf_i \tag{2}$$

$tf_{i,p}$ is the frequency of word $\boldsymbol{t_i}$ in image p. idf_i is defined as (3),

$$idf_i = \log \frac{N}{N_i} \tag{3}$$

N is the total number of training images and N_i is the number of images where word $\boldsymbol{t_i}$ occurs.

The procedures of our approach are as follows:

Training phase:

1. For every image in the training set, we extract the local image features, such as SIFT.
2. Use the clustering algorithm, such as k-means, to cluster the local image features of all the training images to get the words (or the codebook).
3. Calculating the *idf* of every word on the training set by Eq. 3.
4. For every word, adopt cryptographic hashing method to encode it into b-bit fingerprint.

Testing phase:

1. For an image, extract the local image features.
2. Assign the local image features into the cluster based on nearest neighbor rule and compute the *tf-idf* weight w_i of word $\boldsymbol{t_i}$.
3. Encode the image by Eq. 4.

$$B_j = sign(\sum_{i=1}^{c} s_{ij} w_i) \tag{4}$$

c is the size of codebook. s_{ij} is the j-th bit of $\boldsymbol{s_i}$ which is the fingerprint of word $\boldsymbol{t_i}$. B_j is the j-th bit of the simhash code.

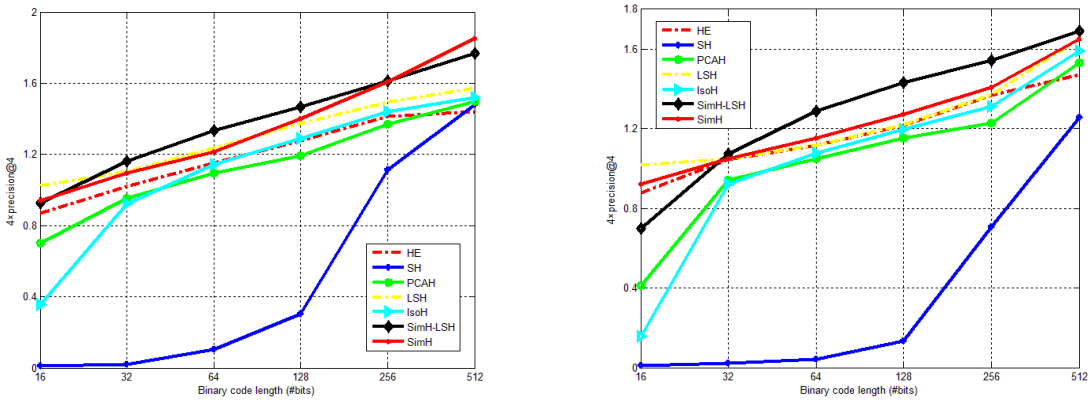
We term the above method SimH. The naive SimH method does not take the correlation of the words into consideration. The fingerprint of similar words will be widely different, which may incur insufficiency especially when the size of codebook is large. So we propose to use "locality-sensitive" hashing methods [3] to encode the words in the fourth step of the training phase to preserve the similarities of the words. For word $\boldsymbol{t_i}$ in the codebook, the j-th bit of its fingerprint $\boldsymbol{s_i}$, s_{ij} can be calculated by Eq. 5.

$$s_{ij} = \begin{cases} -1 & H_j(t_i) < 0 \\ 1 & otherwise \end{cases}, \tag{5}$$

H_j is j-th hash function draw from a Gaussian distribution. Then we can do the same other procedure as SimH. We name this method SimH-LSH.

Experiments

Compared Methods and Datasets. We compare our method with HE, SH, LSH, PCAH, and IsoH on UKB [10] and FLICKR200K (a subset of FLICKR1M [11]). For UKB dataset, we use BoW of 1000-word, and 10000-word respectively to represent the images. The retrieval accuracy is measured in terms of the number of relevant images retrieved in the top 4, i.e. 4×precision@4. For FLICKR200K, we use 100-word BoW representation. We randomly divide the dataset into two part: database (195K) and testing set (5K). And 100K images of the database are used as training set. The performance is measured by mean Average Precision (mAP) and recall. The ground truth is defined by the 1000 nearest neighbors computed by linear scan.



(a) (b)

Figure 1. Experiment results on the UKB dataset: (a) 1000-word BoW, (b) 10000-word BoW.

Results and Analysis. Results on UKB with 1000-word and 10000-word BoW representation are shown in Figure 1. As we can see, our methods achieve better results especially when the bits are longer. SH achieve bad results mainly because that the assumption that the data are multidimensional uniform distributed and that the data have been embedded in a Euclidean space may not hold since the BoW representation is very sparse.

Table 1 shows the results on image set FLICKR200K with 100-word BoW representation. Since HE, SH, PCAH and IsoH can't generate codes longer than data dimensionality while SimH-LSH and SimH can, we do not present the results when codes are longer than 64 bits for HE, SH, PCAH, and IsoH. This image set is very challenging. The content of images differs in thousands of ways and the

Table 1. Experiment on FLICKR200K dataset with 100-word BoW in term of 1000-NN mAP and Recall@1000.

	1000-NN mAP						Recall@1000					
# bits	16	32	64	128	256	512	16	32	64	128	256	512
HE	0.0178	0.0259	0.0322	—	—	—	0.0676	0.0828	0.1194	—	—	—
SH	0.0002	0.0006	0.0015	—	—	—	0.0110	0.0115	0.0187	—	—	—
PCAH	**0.0243**	0.0273	0.0308	—	—	—	**0.0747**	0.0848	0.1139	—	—	—
IsoH	0.0131	0.0285	0.0320	—	—	—	0.0392	**0.0983**	0.1185	—	—	—
LSH	0.0145	0.0225	0.0302	0.0419	0.0617	0.0784	0.0418	0.0773	0.1053	0.1431	0.1746	0.2042
SimH-LSH	0.0213	**0.0287**	**0.0357**	0.0407	0.0499	0.0572	0.0709	0.0935	**0.1238**	0.1429	0.1609	0.1739
SimH	0.0181	0.0261	0.0346	**0.0439**	**0.0656**	**0.0857**	0.0553	0.0830	0.1217	**0.1470**	**0.1785**	**0.2216**

training images cannot reflect the distribution of the whole image set. So all the compared methods achieve poor results. However, among all the methods, our approaches still get better results.

Conclusion

In this work, we propose a new hashing method for image retrieval. It can be intrinsically combined with the popular image representation, BoW. And it is very simple. Based on the naive simhash, we propose to adopt "locality-sensitive" hashing methods like LSH to encode the words. This is especially beneficial when the size of codebook is large. Extensive experiments have demonstrated the effectiveness of our method. Besides, simhash also can be combined with other "locality-sensitive" hashing methods. In the future, we will tackle this issue.

Acknowledgment

This work has been supported by the National Key Technology R&D Program of China under Grant No. 2012BAH04F02, 2012BAH88F02, 2013BAH61F01 and 2013BAH63F01 and the International S&T Cooperation Program of China under Grant No. 2013DFG12980.

References

[1] J. Sivic and A. Zisserman, in: *ICCV*, 2003, p. 1470.

[2] D. Lowe, *International Journal of Computer Vision*, Vol. 66 (2004), p. 91.

[3] M. Datar, N. Immorlica, P. Indyk and V. Mirrokni, in: *Proc. Annu. Symp. Computational Geometry*, 2004, p. 253.

[4] Y. Weiss, A. Torralba, and R. Fergus, in: *NIPS*, 2008, p. 1753.

[5] H. Jégou, M. Douze, and C. Schmid, in *ECCV*, 2008, p. 304.

[6] B. Wang, Z. Li, and M. Li, Technical report, Microsoft Research, 2005.

[7] W. Kong, and W.-J. Li, in *NIPS*, 2012, p. 1655.

[8] Q.-Z. Guo, Z. Zeng, S. Zhang, Y. Zhang, and F. Wang, in: *ICME*, 2013, p. 1.

[9] G. S. Manku, A. Jain, and A. D. Sarma, in *WWW*, 2007, p. 141.

[10] D. Nistér and H. Stewénius, in: *CVPR*, 2006, p. 2161.

[11] M. J. Huiskes, M. S. Lew, in *Proc. Multimedia Information Retrieval*, 2008, p. 39.