Large-scale Image Retrieval with Sparse Binary Projections

Changyi Ma*
218019033@link.cuhk.edu.cn
School of Science and Engineering
*The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China

Chonglin Gu*,†
guchonglin-6@163.com
*The Chinese University of Hong Kong, Shenzhen
†Shenzhen Research Institute of Big Data
Shenzhen, Guangdong, China

Wenye Li*,†
wyli@cuhk.edu.cn
*The Chinese University of Hong Kong, Shenzhen
†Shenzhen Research Institute of Big Data
Shenzhen, Guangdong, China

Shuguang Cui*,†
shuguangcui@cuhk.edu.cn
*The Chinese University of Hong Kong, Shenzhen
†Shenzhen Research Institute of Big Data
Shenzhen, Guangdong, China

ABSTRACT

Inspired by the recent discoveries in neuroscience, the study of the sparse binary projection model started to attract people's attention, shedding new light on image retrieval. Different from the classical work that tries to reduce the dimension of the data for faster retrieval speed, the model projects dense input samples into a higher-dimensional space and outputs sparse binary data representations after *winner-take-all* competition. Following the work along this line, this paper designed a new algorithm which obtains a high-quality sparse binary projection matrix through unsupervised training. Simple as it is, the algorithm reported significantly improved results over the state-of-the-art methods in both search accuracy and retrieval speed in a series of empirical evaluations on large-scale image retrieval tasks, which exhibited its promising potential in industrial applications.

KEYWORDS

Image Retrieval; Sparse Binary Projection; Competitive Learning

ACM Reference Format:
Changyi Ma, Chonglin Gu, Wenye Li, and Shuguang Cui. 2020. Large-scale Image Retrieval with Sparse Binary Projections. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401261

1 INTRODUCTION

The explosive growth of photographs released everyday necessitates the study of information retrieval over a large collection of images [3, 19]. A commonly used *image retrieval* (IR) method is to compare the similarities between the query and the candidate images one by one [16]. However, the computation and the storage demands become expensive when the number of images or the number of features is high. An exhaustive search over the collection consumes nontrivial computation and memory.

To accelerate query speed and reduce storage cost, *approximate nearest neighbor* (ANN) search via hashing is popularly used. With tight theoretical guarantee [8], the *locality sensitive hashing* (LSH) method and its variants [6, 18, 22] perform empirically well. The methods provide a simple way to reduce the storage complexity by trading a controlled amount of representation error for faster processing speed and smaller model sizes.

Very recently, with strong evidences from neuroscience, a sparse binary projection model called the FLY algorithm was designed and started to attract people's attention. The algorithm simulates the fruit fly's olfactory circuits [21, 25]. Instead of performing dimension reduction, the algorithm increases the dimension of the input samples with a random sparse binary projection matrix. After *winner-take-all* (WTA) competition in the output space, a set of sparse binary vectors is obtained as the output representation.

Sparse binary projections bring about two key benefits. One benefit is in similarity search accuracy. In empirical evaluations, it was found that such sparse binary vectors outperformed the dense vectors produced by the LSH method [1]. The other benefit is the significantly improved retrieval speed. As shown in our evaluation (ref. Section 3), comparing images in sparse binary representation can be tens of times faster than in dense representation.

Following the work along this line, we proposed an unsupervised learning algorithm with the explicit addressing of the WTA competition. Instead of residing on random generation of the projection matrix, we sought the optimal binary projection matrix and derived a simple yet effective algorithm. In empirical evaluations, the algorithm reported significantly improved results in image search accuracies and retrieval speed over the state-of-the-art approaches, and hence provided a practical tool of industrial value.

A note on notation. Unless specified otherwise, a lower-cased letter, with or without a subscript, denotes a vector or a scalar. A capital letter denotes a matrix. For example, y_i denotes the i-th element of vector y. Another example, $w_{i\cdot}$ denotes the i-th row vector, $w_{\cdot j}$ denotes the j-th column vector and w_{ij} denotes the (i, j)-th element of matrix W.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401261

2 MODEL

2.1 Proposed Model

We proposed a data-dependent sparse binary projection model [12, 14] for image search. Assume a set of images are given by $X = \{x_{.1}, \cdots, x_{.n}\}$ where each $x_{.m} \in R^d$ $(1 \le m \le n)$ denotes the input representation of an image. Using X as the training data, we hope to obtain a sparse binary projection matrix W that projects the d-dimensional input vectors $\{x_{.1}, \cdots, x_{.n}\}$ to d'-dimensional $(d' \gg d)$ sparse binary output vectors $Y = \{y_{.1}, \cdots, y_{.n}\}$, where each pair of input and output vectors satisfy:

$$y = WTA_k^{d'}(Wx). \tag{1}$$

The function $WTA_k^{d'} : R^{d'} \to \{0,1\}^{d'}$ reflects the *winner-take-all* competition process [15], and satisfies, for all $1 \le i \le d'$,

$$y_i = \begin{cases} 1, & \text{if } (Wx)_i \text{ is within the top-}k \text{ entries of } Wx. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here $k \ll d'$ is referred to as the *hash length* of the projection. To seek the matrix W, we propose to maximize

$$L(W, Y) = \sum_{m=1}^{n} \sum_{i=1}^{d'} \sum_{j=1}^{d'} y_{im} \left(1 - y_{jm}\right) \left(w_{i.} x_{.m} - w_{j.} x_{.m}\right) \tag{3}$$

with respect to W and Y.

Let $\ell_{mij} = y_{im} \left(1 - y_{jm}\right) \left(w_{i.} x_{.m} - w_{j.} x_{.m}\right)$ and we can see:

- If $y_{im} = 1$ and $y_{jm} = 0$, to satisfy the WTA function in Eq. (1), we'd like to have $w_{i.} x_{.m} \ge w_{j.} x_{.m}$ and hence $\ell_{mij} \ge 0$, which contributes to L positively.
- Otherwise, we have $\ell_{mij} = 0$, which does not affect L.

By summing up ℓ_{mij} over all m, i and j, we reach the measure $L(W, Y)$ that quantifies how well a matrix W meets the conditions defined in Eq. (1). Therefore maximizing L with respect to W and Y provides a principled way to seeking the desired projection matrix W and the output representation Y.

In sparse binary projections, we have a similar sparse binary constraint on W, for a given value of c:

$$w_{i.} \in \{0,1\}^d \text{ and } \sum_{j=1}^{d} w_{ij} = c, \text{ for all } 1 \le i \le d'. \tag{4}$$

From the WTA function, we have the constraints on Y:

$$y_{.m} \in \{0,1\}^{d'} \text{ and } \sum_{i=1}^{d'} y_{im} = k, \text{ for all } 1 \le m \le n. \tag{5}$$

2.2 Algorithm

To maximize L with respect to W and Y subject to the constraints defined in Eqs. (4) and (5), we propose an alternating optimization algorithm. Start with a random initialization of W as W^1, and solve the model iteratively. In t-th $(t = 1, 2, \cdots)$ iteration, fix W to W^t and maximize $L\left(W^t, Y\right)$ with respect to Y and get the optimal Y^t. Then fix Y to Y^t and maximize $L\left(W, Y^t\right)$ with respect to W and get the optimal W^{t+1}.

Obviously in each iteration, the optimal Y^t is given by:

$$y_{.m}^t = WTA_k^{d'}\left(W^t x_{.m}\right) \tag{6}$$

for all $1 \le m \le n$.

To get the optimal W^{t+1}, consider the following:

$$\begin{aligned} \max L\left(W, Y^t\right) &\iff \max \sum_{m=1}^{n} \left\{ d' \sum_{i=1}^{d'} y_{im}^t w_{i.} x_{.m} - k \sum_{j=1}^{d'} w_{j.} x_{.m} \right\} \\ &\iff \max \sum_{m=1}^{n} \left[\sum_{i=1}^{d'} y_{im}^t w_{i.} x_{.m} - \frac{k}{d'} \sum_{i=1}^{d'} w_{i.} x_{.m} \right] \\ &\iff \sum_{i=1}^{d'} \max \left\{ w_{i.} \left[\sum_{m=1}^{n} x_{.m} \left(y_{im}^t - \frac{k}{d'} \right) \right] \right\}. \end{aligned} \tag{7}$$

Therefore, the optimal W^{t+1} is given by:

$$w_{i.}^{t+1} = WTA_c^d\left(s_{.i}^t\right) \tag{8}$$

for all $1 \le i \le d'$, where $s_{.i}^t = \sum_{m=1}^{n} x_{.m} \left(y_{im}^t - \frac{k}{d'} \right)$.

Denote by $L^t = L\left(W^t, Y^t\right)$. The sequence $\left\{L^t\right\}$ monotonically increases for $t = 1, 2, \cdots$. Therefore the alternating optimization process is guaranteed to converge and return a local optimal solution when the objective value L^t can't be increased any more.

When the projection matrix W is available, for any testing image, its sparse binary representation (i.e., output representation) can be obtained from (1).

2.3 Complexity Analysis

In each iteration, the proposed training algorithm needs to compute both Y and W successively. In each iteration, computing Y needs $O(cdn + d'd \log k)$ operations by utilizing the sparse structure of W, where $O(cdn)$ are for the multiplication and $O(d'd \log k)$ are for the sorting in Eq. (6). Computing W needs $O(kdn + d'd \log c)$ operations, where $O(kdn)$ are for the summation and $O(d'd \log c)$ are for the sorting in Eq. (8). Therefore, the total complexity is $O((k + c)\, dn + d'd \log(kc))$ per iteration.

The algorithm is highly parallelizable. Each column vector of Y^t and each row vector of W^{t+1} can be computed independently with high parallel efficiency. It is also worth noting that, after simple preprocessing, all training computations only involve simple vector addition and scalar comparison operations, which makes possible to implement the algorithm on customized hardware platform, such as with FPGA technology [17], for further improved efficiency.

The memory requirement is from the storage of the matrices X, Y and W, and the complexity is $O(dn + d'n + d'd)$, which can be reduced to $O(dn + kn + d'c)$ when sparse representation is used. In empirical tasks where the values of d, d', k and c are all fixed. Both the time and the memory complexities grow linearly with n, which provide a scalable solution to real applications.

3 EVALUATION

We conducted experiments on ANN search and IR applications. For simplicity, we referred to our training algorithm as *Sparse Binary Projection* (SBP) which produced sparse binary data representations, and compared it with the following methods:

- LSH (Locality Sensitive Hashing) [2]. The input dimension is reduced with a random dense projection matrix.

- SKLSH (Shift-invariant Kernel Locality Sensitive Hashing) [18]. The projection matrix is generated by random Fourier features for approximating a Gaussian kernel.
- DSH (Density Sensitive Hashing) [7]: The projection matrix is designed to agree with the data distribution.
- CBE (Circulate Binary Embedding) [24]: To speed up projections for high-dimensional data, the elements in a circulant matrix is generated via standard normal distribution.
- ITQ (Iterative Quantization) [23]: The number of non-zero coefficients is restricted in the projection matrix to reduce the parameters and the computation costs.
- FLY (the FLY algorithm) [1]: A sparse binary random projection matrix is used to increase the input dimension, while ensuring the sparsity and binarization of the outputs.

For FLY and SBP, a sparse binary projection matrix (either randomized or trained from the training images) was used to project a d-dimensional input vector to a d'-dimensional output vector with exactly k non-zero elements after WTA competition. Following the work of [1], for LSH and its variants, a projection matrix was used to reduce the data dimension from d to k.

Both ANN and IR were experimented with the following datasets:

- ImageNet [20]: A dataset with 1.2M images of $d = 1,000$ features. We used 1M for training and the rest for testing.
- MNIST [11]: A dataset of 70K images of $d = 784$ pixels. We used 60K for training and the rest for testing.
- CIFAR-10 [9]: A dataset of 60K images of $d = 1,024$ pixels. We used 50K for training and the rest for testing.
- Caltech-101 [5]: A subset of 7,772 images resized to $d = 600$ pixels. We used 6,072 for training and the rest for testing.

Our results were achieved on a server with 44 CPU cores and 384GB memory, with intel MKL as the underlying maths library.

1) Approximate Nearest Neighbor Search

We used ANN search to evaluate how well a hashing algorithm preserved the pairwise similarities within the training set before and after the projection. For each image in the training set, we sought its top-r nearest neighbors based on Euclidean distance (here $r = 100$) in both input and output representations respectively. We recorded the ratio of overlapped top-r neighbors for each image, and used the average ratio as a measure of ANN search accuracy. We set $c = \lfloor 0.1 \times d \rfloor$ and $d' = 2,000$, and reported the mean search accuracy and the standard deviation for all methods after 10 runs of experiments.

Figures 1(a) to 1(d) compares ANN search accuracies on various datasets by different hashing algorithms. It can be seen that, when increasing k from 2 to 32, the mean accuracy of most methods increased accordingly. Among these methods, SBP reported the best results. For example, when $k = 2$, SBP achieved an accuracy of around 12% while all other methods were only around 2% on ImageNet. When $k = 32$, SBP achieved an accuracy of over 50% while all other methods were less than 40% on the same dataset. Similar tendencies were observed on other datasets as well. It can be concluded that the projection matrix obtained from SBP well preserved the pairwise similarities of the input images.

2) Image Retrieval

We used IR to evaluate how well a hashing algorithm preserved the pairwise similarities between the training images and the testing images, which is more tightly related to real applications. For each image in the testing set, we defined the ground-truth of its top-r neighbors as the r training images with the smallest Euclidean distances in the input representation. Then we calculated its top-r neighbors in the output representation, and recorded the overlapping ratio of the two sets of neighbors for the testing image and obtained the accuracy by the average ratio for all testing images. We repeated the experiment for ten runs and reported the mean accuracy and the standard deviation.

Figures 2(a) to 2(d) compares the IR accuracies with fixed $c = \lfloor 0.1 \times d \rfloor$ and various hash length k. From the results we can see that, similarly to the results in ANN search, SBP was more accurate than FLY, LSH and its variants in 100-nearest neighbor IR. The improvement was especially significant for small k. Take ImageNet as an example, when the hash length $k = 2$, SBP achieved an accuracy of over 12%, while all other methods only had an accuracy around 2%. Besides, SBP had a much smaller standard deviation of accuracies, which further justified its performance stability.

3) Training and Retrieval Speed

To evaluate the running speed on large-scale IR tasks, we randomly generated a database of 10M images with sparse binary and dense representations respectively. For each query image, the IR computations are from two phases:

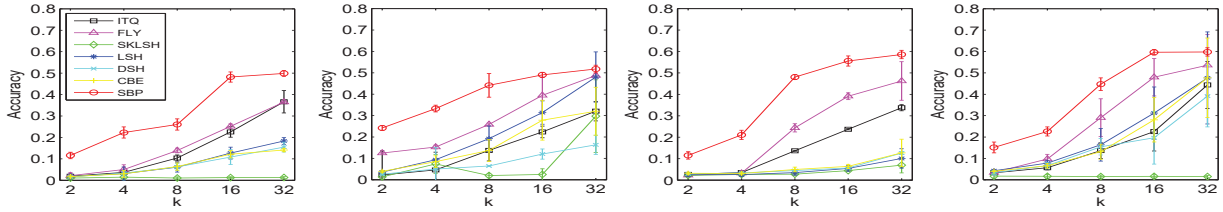
- Projection: compute the hashing vector for the image;
- Retrieval: compute and compare the distance between the query image with all candidate images in the database.

We compared the projection and retrieval time over this database by the SBP and the LSH methods, shown in figures 3(a) and 3(b). Although SBP took longer time than LSH in the projection phase, it was much faster in the retrieval phase. Summing up the projection time and the retrieval time, SBP took about only 0.5 seconds to search for an image over such a large database, which was over 20 times faster than LSH on various hash length k. Our results verified the benefit of sparse binary representations which makes possible to apply faster logical operations instead of arithmetic operations in distance or similarity calculation.

4 CONCLUSION

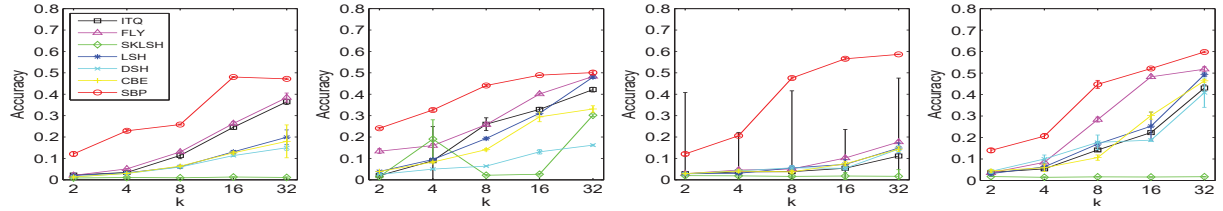
In this paper, we developed an unsupervised learning algorithm to compute sparse binary projection matrix with high quality. Simple as it is, the algorithm reported significantly improved results, in both accuracy and speed, over state-of-the-art projection-based methods in large-scale IR tasks. Our result is consistent with the observation in the work of [4, 10, 13] which showed that high-dimensional features from large-scale data can improve the performance of various pattern recognition tasks.

In addition to the improvement in search accuracies, a key benefit of the proposed algorithm is that it has fast retrieval speed and scalable storage requirement with the sparse binary representation. The proposed algorithm supports parallel execution with high efficiency. For ongoing work, we are investigating both parallel computing and customized hardware to further accelerate the running speed of the algorithm and fully demonstrate its promising potential in industrial applications.



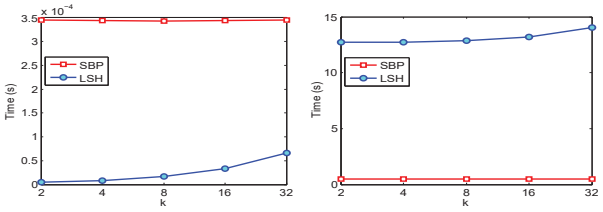
(a) ImageNet (b) MNIST (c) CIFAR (d) Caltech-101

Figure 1: Comparisons of ANN search accuracy on various datasets



(a) ImageNet (b) MNIST (c) CIFAR (d) Caltech-101

Figure 2: Comparisons of IR accuracy on various datasets



(a) Projection Speed (b) Retrieval Speed

Figure 3: Running time on 10M images ($d' = 2,000$)

ACKNOWLEDGMENT

Correspondence should be addressed to Wenye Li at The Chinese University of Hong Kong, Shenzhen. Wenye's work was supported by Key-Area Research and Development Program of Guangdong Province (2018B030338001), The Science and Technology Innovation Committee of Shenzhen (JCYJ20170306141038939, KQJSCX20170728162302784, JCYJ20170410172341657), and Guangdong Introducing Innovative and Entrepreneurial Teams Fund (2017ZT07X152), China. Chonglin's work was supported by China Postdoctoral Science Foundation (2019M650148), and Guangdong Basic and Applied Basic Research Foundation (2019A1515110214).

REFERENCES

[1] S. Dasgupta, C. Stevens, and S. Navlakha. 2017. A neural algorithm for a fundamental computing problem. *Science* 358, 6364 (2017), 793–796.
[2] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *20th Annual Symposium on Computational Geometry*. 253–262.
[3] R. Datta, D. Joshi, J. Li, and J. Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. *Comput. Surveys* 40, 2 (2008), 1–60.
[4] J. Donahue, Y. Jia, O. Vinyals, et al. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*. 647–655.
[5] L. Fei-Fei, R. Fergus, and P. Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 178–178.
[6] A. Gionis, P. Indyk, R. Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *25th International Conference on Very Large Data Bases*, Vol. 99. 518–529.
[7] Z. Jin, C. Li, Y. Lin, and D. Cai. 2013. Density sensitive hashing. *IEEE Transactions on Cybernetics* 44, 8 (2013), 1362–1371.
[8] W. Johnson and J. Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics* 26, 189-206 (1984), 1.
[9] A Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Technical Report.
[10] A. Krizhevsky, I. Sutskever, and G. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[12] W. Li. 2019. Modeling winner-take-all competition in sparse binary projections. *CoRR* abs/1907.11959 (2019).
[13] W. Li and S. Hao. 2019. Sparse Lifting of Dense Vectors: Unifying Word and Sentence Representations. *CoRR* abs/1911.01625 (2019).
[14] W. Li, J. Mao, Y. Zhang, and S. Cui. 2018. Fast similarity search via optimal sparse lifting. In *Advances in Neural Information Processing Systems*. 176–184.
[15] W. Maass. 2000. On the computational power of winner-take-all. *Neural Computation* 12, 11 (2000), 2519–2535.
[16] S. McDonald and J. Tait. 2003. Search strategies in content-based image retrieval. In *26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. 80–87.
[17] A. Omondi and J. Rajapakse. 2006. *FPGA Implementations of Neural Networks*. Vol. 365. Springer.
[18] M. Raginsky and S. Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*. 1509–1517.
[19] Y. Rui, T. Huang, and S. Chang. 1999. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation* 10, 1 (1999), 39–62.
[20] O. Russakovsky, J. Deng, H. Su, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.
[21] C. Stevens. 2015. What the fly's nose tells the fly's brain. *Proceedings of the National Academy of Sciences* 112, 30 (2015), 9460–9465.
[22] J. Wang, S. Kumar, and S. Chang. 2010. Semi-supervised hashing for scalable image retrieval. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 3424–3431.
[23] Y. Xia, K. He, P. Kohli, and J. Sun. 2015. Sparse projections for high-dimensional binary codes. In *2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 3332–3339.
[24] F. Yu, S. Kumar, Y. Gong, and S. Chang. 2014. Circulant binary embedding. In *International Conference on Machine Learning*. 946–954.
[25] Z. Zheng, S. Lauritzen, E. Perlman, et al. 2018. A complete electron microscopy volume of the brain of adult Drosophila melanogaster. *Cell* 174, 3 (2018), 730–743.