EFFICIENT BINARY CODES FOR EXTREMELY HIGH-DIMENSIONAL DATA

Tsung-Yu Lin, Tyng-Luh Liu

Institute of Information Science, Academia Sinica, Taiwan

ABSTRACT

Recent advances in tackling large-scale computer vision problems have supported the use of an extremely high-dimensional descriptor to encode the image data. Under such a setting, we focus on how to efficiently carry out similarity search via employing binary codes. Observe that most of the popular high-dimensional descriptors induce feature vectors that have an implicit 2-D structure. We exploit this property to reduce the computation cost and high complexity. Specifically, our method generalizes the Iterative Quantization (ITQ) framework to handle extremely high-dimensional data in two steps. First, we restrict the dimensionality-reduction projection to a block-diagonal form and decide it by independently solving several moderate-size PCA sub-problems. Second, we replace the full rotation in ITQ with a bilinear rotation to improve the efficiency both in training and testing. Our experimental results on a large-scale dataset and comparisons with a state-of-the-art technique are promising.

***Index Terms*—** Binary code, hashing, similarity search

1. INTRODUCTION

Identifying images (or image patches) of interest from a given collection of candidates is fundamental to many computer vision techniques. As the scale of image database grows enormously, how to design methods to efficiently and satisfactorily carry out the retrieval task has attracted much attention. Among the many research efforts, quite a number of hashing schemes to yield *binary codes* have been proposed to approximate nearest-neighbor search, *e.g.*, [1, 2, 3, 4]. Nevertheless, while most of these approaches are supported by theoretical foundation and demonstrated with acceptable performances over popular benchmark datasets, they often target at dealing with low-dimensional feature vectors such as 512-D GIST. It is generally hard to directly use them to accommodate the case that each image is represented by an extremely high-dimensional (say, more than 100K dimensions) feature vector. The predicament is mostly due to two practical issues. First, the extremely high-dimensional setting would cause the underlying dimensionality reduction technique to become unfeasible or require excessive time and computing resources in learning the mapping. Second, even when the difficulties in training could be overcome, the resulting hashing scheme is simply computationally too expensive to be applied to computing the binary code of a query image. On the other hand, recent advances in vision research have revealed that for large-scale problems like fine-grained level object categorization over ImageNet [5], adopting an extremely high-dimensional descriptor is preferable [6, 7]. The promising results suggest that extending binary codes for high-dimensional data would very likely expand the usefulness of the technique in handling large-scale and complicated computer vision problems and applications.

We explore the implicit 2-D structure embedded in many of the popular high-dimensional descriptors, *e.g.*, Fisher Vectors (FV) [8], Vectors of Locally Aggregated Descriptors (VLAD) [7] and Locality constrained Linear Codes (LLC) [9], and decompose it into sub-structures of smaller dimensions that can be independently tackled. Our approach is motivated by Product Quantization (PQ) [10], which generates a large number of quantization centroids by regularly subdividing the feature space. Specifically, like ITQ [1], the proposed method needs to compute a dimensionality-reduction mapping and a rotation matrix for yielding binary codes. With the feature decomposition, the mapping can be represented by a very sparse block diagonal matrix so that memory storage and computation time for generating a binary code can be significantly reduced. Indeed the resulting mapping can be constructed by simultaneously solving a PCA problem with respect to each decomposed feature space. To decide the rotation matrix, we restrict it to the bilinear rotation as described in [11]. Our experiments show that the retrieval results by our method empirically well approximate those by ITQ but with substantial improvement in efficiency.

2. RELATED WORK

We aim to introduce an unsupervised technique to generate binary codes for extremely high-dimensional data. The literature survey thus focuses mainly on relevant work about unsupervised binary coding and recent techniques for constructing and handling high-dimensional data/descriptors.

To efficiently perform approximate similarity search, Locality Sensitive Hashing (LSH) by Gionis *et al.* [12] relies on random projections, each of which ensures that the probability of hashing collision is closely related to the distance between two points. In [1], Gong and Lazebnik propose the

978-1-4799-5751-4/14/$31.00 ©2014 IEEE

Iterative Quantization (ITQ) framework that instead of using random projections, the method first reduces data dimensions to the desired code length by a PCA projection and then decides a rotation matrix to minimize the quantization loss of binary codes. While ITQ has been shown to be effective for handling large-scale data described by an image descriptor of moderate dimensions, the technique does not generalize well to high-dimensional data. It would not only take considerable memory space to store the PCA projection of extremely high dimensions but also demand an extensive computation in performing dimensionality reduction. More critically, computing the PCA projection and optimizing the rotation matrix in training would become unfeasible. To address these issues, Gong *et al.* [11] further propose to use a bilinear projection to replace the lower-dimensional projection followed by rotation technique of ITQ. However, the method completely skips the PCA step and achieves dimensionality reduction only by simplifying a bilinear rotation.

As computer vision techniques move in progress with new challenges emerging constantly, the effectiveness of high-dimensional features has been demonstrated in various applications, including image classification [6, 7] and face verification [13]. The often-used high-dimensional image descriptors, including LLC [9], FV [8] and VLAD [7], are proposed to surpass the limitation of Bag-of-Words representation, which considers only the 0th-order statistics of features. Derived by approximately solving locality-constrained sparse representations and pooling over a spatial pyramid, the LLC feature vector has a dimension specified by the product between the size of the dictionary and the number of spatial bins. Both VLAD and FV can encode 1st-order statistics (optionally, 2nd-order for FV). The dimension of FV is twice of the dimension of VLAD descriptor, which is the size of the dictionary multiplied by the dimension of the local image feature. In view of how their dimension is computed, it implies that the aforementioned high-dimensional descriptors have a nature 2-D representation in matrix form, which can be conveniently exploited by bilinear mappings [11].

Adopting a high-dimensional feature representation unavoidably leads to high cost both in training and testing phases. Hence, how to craftily utilize such a descriptor is pivotal in advancing the performance. To this end, PQ [10] decomposes the vector space and independently quantizes each subspace to yield quantization centroids. Sánchez and Perronnin [14] further suggest that PQ is suited to balancing classification accuracy, computational cost and storage cost for high-dimensional data. With regard to dimensionality reduction, Chen *et al.* [13] approximate a lower-dimensional projection by sparse regression, which can reduce, without sacrificing accuracy, computational cost and memory usage. The main idea of our method is relevant to these techniques in that the proposed approach to overcoming the challenging computation cost critically relies on decomposing the high-dimensional feature space and the mappings.

3. EFFICIENT BINARY CODES

The crux of our method consists in how to make use of the implicit 2-D structure of a high-dimensional descriptor so that computation and resource demanding optimization problems are reduced to those that can be handled more efficiently. In addition, since our approach extends ITQ to handle extremely high-dimensional data, it is convenient to lay out its formulation to better understand what prevent ITQ from being efficient and how we could resolve these difficulties.

3.1. ITQ

ITQ is an unsupervised technique for computing binary codes. It comprises two main steps to achieve the task, namely, to compute a lower-dimensional projection and to optimize a rotation matrix.

Given a set of data points $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$, we form the data matrix $D \in \mathbb{R}^{d\times n}$ with $\mathbf{x}_i$ being the ith column. ITQ is to compute the binary codes $B \in \{-1,1\}^{c\times n}$ where its ith column, denoted as $\mathbf{b}_i$, is the c-bit ($c < d$) binary code of $\mathbf{x}_i$. The algorithm begins with finding c projection directions $\{\mathbf{w}_k \in \mathbb{R}^d\}_{k=1}^c$ such that the variance of each bit is maximized and the bits are pairwise uncorrelated. We use $W \in \mathbb{R}^{d\times c}$ to express the resulting dimensionality-reduction projection where $\mathbf{w}_k$ is the kth column. It turns out that finding the optimal W can be casted as solving a PCA problem:

$$W^* = \arg\max_W \sum_{k=1}^{c} \mathbb{E}(\|\mathbf{w}_k^T \mathbf{x}\|_2^2), \quad W^T W = I. \tag{1}$$

After using PCA to carry out the dimensionality reduction, the next important step of ITQ is to obtain a rotation matrix $R \in \mathbb{R}^{c\times c}$, which will be applied to the projected data so that the quantization loss of the resulting binary codes is minimized. We have

$$\{B^*, R^*\} = \arg\min_{B,R} \|B - R^T W^{*T} D\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The optimization problem (2) can be effectively solved as described in [1] and the binary codes are given by

$$B^* = \operatorname{sgn}(R^{*T} W^{*T} D). \tag{3}$$

3.2. Our Method

With (1)-(3), it can be readily inferred that ITQ would degrade into an inefficient scheme when the feature dimension d becomes extremely large. To deal with the high complexity, Gong *et al.* [11] propose to omit the PCA dimensionality reduction in that solving (1) is too computationally expensive. While the tactic is handy in avoiding a challenging extremely large-scale PCA problem, it nevertheless loses the nice property of bit-wise balance embodied in ITQ.

We instead choose to approximately and efficiently solve (1) by taking account of the implicit 2-D structures of the extremely high-dimensional descriptors mentioned in Section 2. Specifically, we assume that the high dimension is expressed by $d = d_r \times d_c$. Take, for example, the case of using a VLAD descriptor constructed based on a dictionary of 1024 atoms, each of which is described by SIFT. Then we have $d_r = 128$ and $d_c = 1024$. To be able to manage the computation loading of (1), we restrict the dimensionality-reduction projection W to a sparse block-diagonal matrix. That is,

$$W = \begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_{d_c} \end{bmatrix} \in \mathbb{R}^{d \times c} \tag{4}$$

where nonzero elements in W appear only at the diagonal blocks $W_j \in \mathbb{R}^{d_r \times \hat{c}}$, $j = 1, \ldots, d_c$ and $\hat{c} = c/d_c$. Correspondingly, we consider decomposing each extremely high-dimensional data point $\mathbf{x} \in \mathbb{R}^d$ into d_c segments and represent its jth segment by $\mathbf{x}(j) \in \mathbb{R}^{d_r}$. The PCA problem in (1) can then be reduced to

$$\begin{aligned} & W^* = \arg\max_W \sum_{j=1}^{d_c} \sum_{k=1}^{\hat{c}} \mathbb{E}(\|\mathbf{w}_{jk}^T \mathbf{x}(j)\|_2^2), \\ & W_j^T W_j = I \in \mathbb{R}^{\hat{c} \times \hat{c}}, \quad \forall j \in \{1, \ldots, d_c\}, \end{aligned} \tag{5}$$

where $\mathbf{w}_{jk}$ is the kth column of W_j. The new PCA problem posed in (5) implies that it can be decomposed into d_c tasks, each of which is now a smaller-size of PCA problem and can be solved efficiently and independently. Finally, we denote the mapping of each data point by W^* as

$$\mathbf{x}_i \in \mathbb{R}^d \longmapsto \mathbf{y}_i = W^{*T} \mathbf{x}_i \in \mathbb{R}^c, \quad \forall i \in \{1, \ldots, n\}. \tag{6}$$

Analogous to ITQ, the remaining task is to find the optimal rotation matrix $R^* \in \mathbb{R}^{c \times c}$ that minimizes the quantization loss of the resulting c-bit binary codes. Since the length of a binary code could be quite large, say, more than 10000, directly solving (2) is still impractical. Inspired by [11], we limit R to a bilinear rotation to alleviate the computation burden. That is, we can then write $R = R_2 \otimes R_1$ where compared with R, both $R_1 \in \mathbb{R}^{\hat{c} \times \hat{c}}$ and $R_2 \in \mathbb{R}^{d_c \times d_c}$ are rotations of a much smaller size, and $\otimes$ denotes the Kronecker product. (Recall that $c = \hat{c} \times d_c$.) We follow the optimization technique in [11] to compute the optimal R_1^* and R_2^*. Let $Y_i \in \mathbb{R}^{\hat{c} \times d_c}$ be the 2-D matrix form of $\mathbf{y}_i$. Then the c-bit binary code of $\mathbf{x}_i, i = 1, \ldots, n$ can be obtained by

$$\mathbf{b}_i = \text{vec}(\text{sgn}(R_1^{*T} Y_i R_2^*)) \tag{7}$$

where the notation $\text{vec}(\cdot)$ reshapes a matrix into a vector by column-wise stacking.

4. EXPERIMENTS AND DISCUSSIONS

We evaluate the effectiveness of our approach on a large-scale dataset. The experiments are to verify our claim that retaining the dimensionality-reduction PCA step is crucial in generalizing ITQ to deal with extremely high-dimensional data. Our experimental results show that the proposed method empirically well approximates ITQ when using a sufficiently long code length, a case that ITQ cannot be efficiently applied.

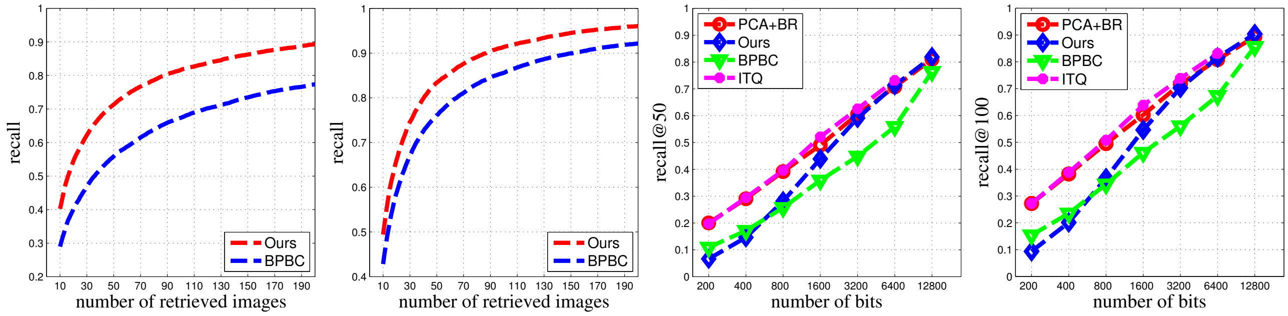
4.1. Evaluation Protocols

We test our method on ILSVRC2010, which is a subset of ImageNet, and conduct performance comparisons with BPBC [11], a state-of-the-art algorithm that computes binary codes for high-dimensional data. The dataset includes 1.2M images over 1000 categories. We randomly select 30K images to learn the PCA projection W^* in (5) and the bilinear rotation $R^* = R_2^* \otimes R_1^*$ for our method. For the sake of comparison, we adopt VLAD and LLC image descriptors, which are also used in BPBC. We download from the ImageNet website the public SIFT features which are densely extracted at three different scales. To construct the VLAD descriptor, we form 200 clusters and assign each SIFT feature to one of the clusters to represent an image. The resulting dimension is $d = d_r \times d_c = 128 \times 200 = 25600$. As suggested in [15], we normalize VLAD feature vectors by intra-normalization followed by L2-normalization. For the LLC descriptor, we construct a dictionary of size 5000 and aggregate LLC representation using a three-level spatial pyramid and max pooling. The aggregated feature vector is further processed by zero-centering and L2-normalization. The dimension of the resulting LLC is $d = d_r \times d_c = 5000 \times 21 = 105000$.

To evaluate the performance, we randomly sample 1000 images, which are not used in training, as query images. The ground-truth nearest neighbors are defined as the top 10 nearest neighbors measured by Euclidean distance. The recall is computed by the top k retrieved images based on Hamming distance. For a fair comparison, BPBC is trained with the exact same data. In addition, when using VLAD, we also look into how well our method can approximate ITQ if the PCA problem in (1) can be solved exactly. For all methods, we use the same random splits of training and testing data.

4.2. Retrieval Results

Figures 1(a) and 1(b) show the performance comparisons between our method and BPBC when the data are described by VLAD. The recall of 10NN is evaluated with different numbers of top k returns. The results indicate that our method, when encoded with sufficient number of bits such as 6400 or 12800, can reasonably preserve the distance measures in the high-dimensional ($d = 25600$) space. It outperforms BPBC by more than 10% when using a moderate code size like 6400



(a) $c = 6400$ (b) $c = 12800$ (c) $k = 50$ (d) $k = 100$

Fig. 1. [25600-D VLAD] Retrieval performance is measured by averaged recall of the top k returned images for each query. In (a) and (b), the curves of recall vs. value of k are plotted when code length c is respectively set to 6400 and 12800 for both methods. In (c) and (d), the curves of recall vs. code size c are plotted when k is respectively set to 50 and 100 for all methods.

bits. The effectiveness justifies that retaining the PCA projection step is useful and enables our method to require less bits to preserve the neighborhood relations than BPBC.

We next compare our method with ITQ to see the effect of solving the PCA problem by assuming a block-diagonal projection W. Note that the results by ITQ are plotted only up to 6400 bits in Figures 1(c) and 1(d) since using ITQ to yield binary codes of 12800 bits is too computationally expensive to complete. Instead, we replace the computation of a full rotation in ITQ with a bilinear rotation and term the technique as PCA+BR. Thus, there are totally four methods to be investigated, including ITQ, PCA+BR, BPBC and ours. In Figures 1(c) and 1(d), we observe that the red curves are almost aligned with the magenta ones, indicating that when the used descriptor (like VLAD) displays 2-D structure, bilinear rotations are suitable for yielding binary codes. Stacking up the recall results by ITQ, PCA+BR and ours, they are comparable when the code size is larger than 3200. On the other hand, the short-code performance by our method drops significantly and even falls below that by BPBC. The phenomenon is mainly caused by our simplification of the PCA projection that would lose too much information when the code length is short. This is not harmful as in practice long binary codes are needed to carry out approximate similarity search for extremely high-dimensional data. It can also be observed from Figures 1(c) and 1(d) that long codes are necessary for the four methods to perform satisfactorily. To demonstrate the scalability of our method, we reduce the data dimension by half as in [11] to obtain binary codes of 52500 bits for LLC feature. Figure 2 shows the results of recall versus number of retrieved images. As BPBC works quite well with LLC, our method could still further improve the performance.

On computational efficiency, we report the required time by ITQ and our method in learning the PCA projection and applying the projection in testing, respectively. Notice that when using the LLC descriptor, the results by ITQ cannot be obtained due to the extremely heavy computation cost (after running for more than one week) and hence not reported. All running time in Table 1 is computed on a PC with 2-core 2.4GHz 24GB RAM in Matlab implementation. The results show that our method substantially reduces the running time both in training and testing phases. Especially with VLAD, our method adds only negligible cost on testing while gaining significant improvements on retrieval results.

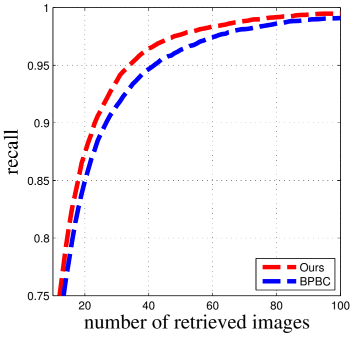


Fig. 2. [105000-D LLC] Retrieval performance is measured by averaged recall of the top k returned images for each query. The code length c is set to 52500 for both methods.

Table 1. Time for learning a PCA projection and average time for applying dimensional reduction in coding step. The code size is 12800 and 52500 for VLAD and LLC respectively.

	ITQ		Ours	
	training	testing	training	testing
VLAD	∼ 20 hrs	1.31 secs	46.46 secs	0.007 secs
LLC	*	*	∼ 5 hrs	1.15 secs

Acknowledgement: We thank the reviewers for their valuable comments. This work was supported in part by MOST grant 102-2221-E-001-021-MY3.

5. REFERENCES

[1] Yunchao Gong and Svetlana Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2011, pp. 817–824.

[2] Kaiming He, Fang We, and Jian Sun, “K-means hashing: an affinity-preserving quantization method for learing binary compact codes,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013.

[3] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang, “Hashing with graphs,” in *Proc. Int. Conf. on Machine Learning*, 2011, pp. 1–8.

[4] Yair Weiss, Antonio Torralba, and Robert Fergus, “Spectral hashing,” in *Advances in Neural Information Processing Systems*, 2008, pp. 1753–1760.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[6] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, “Improving the Fisher kernel for large-scale image classification,” in *Proc. Euro. Conf. on Computer Vision*, pp. 143–156. Springer, 2010.

[7] Herv Jgou, Matthijs Douze, Cordelia Schmid, and Patrick Prez, “Aggregating local descriptors into a compact image representation,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[8] Florent Perronnin and Christopher Dance, “Fisher kernels on visual vocabularies for image categorization,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[9] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong, “Locality-constrained linear coding for image classification,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.

[10] Herve Jegou, Matthijs Douze, and Cordelia Schmid, “Product quantization for nearest neighbor search,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[11] Yunchao Gong, Sanjiv Kumar, Henry A. Rowley, and Svetlana Lazebnik, “Learning binary codes for high-dimensional data using bilinear projections,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013, pp. 484–491.

[12] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, “Similarity search in high dimensions via hashing,” in *Proc. Int. Conf. on Very Large Data Bases*, 1999, pp. 518–529.

[13] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun, “Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013, pp. 3025–3032.

[14] Jorge Sánchez and Florent Perronnin, “High-dimensional signature compression for large-scale image classification,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2011, pp. 1665–1672.

[15] Relja Arandjelovic and Andrew Zisserman, “All about VLAD,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.