Sparse Composite Quantization

Ting Zhang[1], Guo-Jun Qi[2], Jinhui Tang[3], Jingdong Wang[4]
[1]University of Science and Technology of China. [2]University of Central Florida. [3]Nanjing University of Science and Technology. [4]Microsoft Research Beijing.

Motivation. This paper concerns the quantization algorithms for approximate nearest neighbor (ANN) search with high-dimensional data. The recently proposed composite quantization approach [4] introduces a new framework generalizing product quantization [2] and its extensions with optimized space partitions, Cartesian k-means [3]. The acceleration stems from the ability of efficiently computing the distance between a query and a database vector by looking up a distance table computed in the online query stage before scanning the whole database.

Previous approaches mainly focus on improving the accuracy of the distance approximation, while pay little attention to reducing the cost of the online distance table construction. The construction cost becomes a significant contribution to the ANN search cost in real applications, e.g. reordering the candidates retrieved from inverted index when handling very large databases, which is the application we mainly focus on. To handle this issue, we introduce a novel approach, sparse composite quantization, which generalizes the composite quantization approach by allowing the words in the dictionaries to be sparse.

Formulation. Given a database containing N data vectors $\mathcal{X}=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\}$ of dimension D, the proposed approach 1) learns M dictionaries $\{\mathbf{C}_1,\mathbf{C}_2,\cdots,\mathbf{C}_M\}$, with each containing a few number of (assumed to be K without loss of generality) D-dimensional vectors (words) denoted by $\mathbf{C}_m=[\mathbf{c}_{m1}\ \mathbf{c}_{m2}\cdots\mathbf{c}_{mK}]$; 2) approximates each database vector using a composite vector $\bar{\mathbf{x}}=\sum_{m=1}^{M}\mathbf{c}_{mk_m}$, where $\mathbf{c}_{mk_m}$ is the k_mth vector in the mth dictionary $\mathbf{C}_m$; and 3) represents the database vector $\mathbf{x}$ using a short code of length $M\log K$, a sequence of indices $(k_1,k_2,\cdots,k_M)$.

The distance between a query vector $\mathbf{q}$ and a database vector $\mathbf{x}$ is approximated as below,

$$\|\mathbf{q}-\mathbf{x}\|_2^2\approx\|\mathbf{q}-\bar{\mathbf{x}}\|_2^2=\|\mathbf{q}-\sum_{m=1}^{M}\mathbf{c}_{mk_m}\|_2^2 \tag{1}$$
$$=\sum_{m=1}^{M}\|\mathbf{q}-\mathbf{c}_{mk_m}\|_2^2-(M-1)\|\mathbf{q}\|_2^2+\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\mathbf{c}_{ik_i}^{\top}\mathbf{c}_{jk_j}.$$

By introducing a constant constraint on inter-dictionary- element-product ($\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\mathbf{c}_{ik_i}^{\top}\mathbf{c}_{jk_j}=$ constant) in composite quantization [4], the approximate distance computation is simplified to only compute the first term $\sum_{m=1}^{M}\|\mathbf{q}-\mathbf{c}_{mk_m}\|_2^2$ to search the nearest neighbors, which furthermore is accelerated by looking up a distance table storing the distances between the query and the words of all the M dictionaries.

The distance table construction consists of MK distance computations: $\{\|\mathbf{q}-\mathbf{c}_{mk}\|_2^2\mid m=1,2,\cdots,M;k=1,2,\cdots,K\}$, which takes $O(MKD)$ time. Compared with the approximate distance computation cost over all N data vectors, $O(MN)$, the distance table cost is negligible if $N\gg KD$. Nonetheless, considering a real application, where we reorder the candidates retrieved from inverted index [1] for 128-dimensional SIFT features with a typical setting, $K=256$ and $M=8$, the number of retrieved candidates N (e.g., $=100,000$) is not much greater than (about 3 times) KD ($\approx 32,000$) and such cost of distance table computation becomes significant and non-negligible.

Our idea, to accelerate the distance table construction, is inspired by the property: computing the Euclidean distance between two vectors, $\|\mathbf{q}-\mathbf{c}\|_2^2$, can be accelerated if the vector $\mathbf{c}$ is very sparse and $\|\mathbf{q}\|_2^2$ is already known, and the complexity in general becomes $O(\|\mathbf{c}\|_0)$, i.e., linear in the number of non-zero entries. Rather than enforcing each dictionary word being sparse (i.e., let each $\|\mathbf{c}_{mk}\|_0$ be $\frac{D}{M}$, into which we will show that product quantization can be cast), we impose a global sparsity constraint over all the dictionary words together, e.g., $\sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{c}_{mk}\|_0\leqslant KD$.

To this end, the objective is to minimize the vector approximation error, with the constant inter-dictionary- element-product constraint and the sparsity regularization. We formally formulate the objective function as follows,

$$\min_{\{\mathbf{C}_m\},\{\mathbf{y}_n\},\xi}\ \sum_{n=1}^{N}\|\mathbf{x}_n-[\mathbf{C}_1\mathbf{C}_2\cdots\mathbf{C}_M]\mathbf{y}_n\|_2^2 \tag{2}$$
$$\text{s.t.}\ \sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{c}_{mk}\|_0\leqslant S \tag{3}$$
$$\sum_{m=1}^{M}\sum_{m'=1,m'\neq m}^{M}\mathbf{y}_{nm}^{\top}\mathbf{C}_m^{\top}\mathbf{C}_{m'}\mathbf{y}_{nm'}=\xi \tag{4}$$
$$n=1,2,\cdots,N. \tag{5}$$

Here S is a positive constant indicating the sparsity degree. $\mathbf{y}_n=[\mathbf{y}_{n1}^{\top}\mathbf{y}_{n2}^{\top}\cdots\mathbf{y}_{nM}^{\top}]^{\top}$, and $\mathbf{y}_{nm}$ is a binary vector with only one entry valued as 1 and all others as 0. It is meant that the kth word of the dictionary $\mathbf{C}_m$ is selected to form the approximation of $\mathbf{x}_n$ if the kth entry y_{nmk} is equal to 1. $[\mathbf{C}_1\mathbf{C}_2\cdots\mathbf{C}_M]\mathbf{y}_n$ is equivalent to the definition of the approximation $\bar{\mathbf{x}}=\sum_{m=1}^{M}\mathbf{c}_{mk_m}$.

Experiments. Optimization algorithm of the above formulation is described in the paper, as are the details of the experiments, which include three parts. First, we demonstrate the performance over three middle and large scale datasets (MNIST, 1M SIFT and 1M Tiny). Second, we show the performance over a very large scale dataset (1B SIFT) using the inverted multi-index that has the ability of efficiently retrieving the candidates from a large number of inverted lists. Last, we apply our approach to the object retrieval problem over the INRIA Holidays dataset and the UKBench dataset.

As shown in the paper, the proposed approach constructs the distance lookup table as fast as the most efficient algorithm, product quantization, while achieving a higher accuracy than product quantization, and also a similar or even higher accuracy when compared to Cartesian k-means and composite quantization. On the challenging search task with 1 billion SIFT vectors, the proposed approach outperforms the state-of-the-art algorithms in terms of both search efficiency and search accuracy. In particular, compared with composite quantization that achieves the highest recall performance, the proposed approach has successfully reduced the search time by at least 7% up to 36% in many cases, while the reduction of recall is limited within less than 3%.

Conclusion. In conclusion, the proposed approach generalizes the composite quantization by sparsifying the elements in the dictionaries resulting in the efficient construction of a distance lookup table storing distances between the query and the dictionary words. The empirical results show that our approach achieves superior performance in terms of search accuracy and search efficiency.

[1] Artem Babenko and Victor S. Lempitsky. The inverted multi-index. In *CVPR*, pages 3069–3076, 2012.

[2] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.

[3] Mohammad Norouzi and David J. Fleet. Cartesian k-means. In *CVPR*, pages 3017–3024, 2013.

[4] Ting Zhang, Chao Du, and Jingdong Wang. Composite quantization for approximate nearest neighbor search. In *ICML (2)*, pages 838–846, 2014.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.