Sub-Selective Quantization for Learning Binary Codes in Large-Scale Image Search

Yeqing Li, *Member, IEEE*, Wei Liu, *Member, IEEE*, and Junzhou Huang, *Member, IEEE*

Abstract—Recently with the explosive growth of visual content on the Internet, large-scale image search has attracted intensive attention. It has been shown that mapping high-dimensional image descriptors to compact binary codes can lead to considerable efficiency gains in both storage and performing similarity computation of images. However, most existing methods still suffer from expensive training devoted to large-scale binary code learning. To address this issue, we propose a sub-selection based matrix manipulation algorithm, which can significantly reduce the computational cost of code learning. As case studies, we apply the sub-selection algorithm to several popular quantization techniques including cases using linear and nonlinear mappings. Crucially, we can justify the resulting sub-selective quantization by proving its theoretic properties. Extensive experiments are carried out on three image benchmarks with up to one million samples, corroborating the efficacy of the sub-selective quantization method in terms of image retrieval.

Index Terms—Feature quantization, dimensionality reduction, image search, image retrieval, large-scale machine learning

1 Introduction

The explosive growth of the Internet has brought about great opportunities as well as challenges to information technology research. The number of webpages on the World Wide Web has surpassed the trillion level. More than 5 billion images have been uploaded to *Flickr* with an uploading rate of over 3,000 images per minute. On the other hand, roughly more than 100 hours of videos are uploaded to *YouTube* per minute. The rapid growth of the size of data boosts the need of scalable techniques for handling large-scale data sets. One of the most interesting and important problem is Content-Based Image Retrieval (CBIR) on large-scale data sets. This problem can be modeled as similarity search under some predefined distance metrics.

Similarity search has served as a fundamental technique used in many vision-related applications including object recognition [1], [2], image retrieval [3], [4], image matching [5], [6], etc. It is a related problem of nearest neighbor search (NNS) [7]. The naive solution to NNS is exhaustively comparing a query point with each sample in the database. Suppose that there are N data points in a database. The time complexity will be linear $\mathcal{O}(N)$, which is impractical for many real-world applications on large-scale data sets. Another challenge of performing machine learning and data mining algorithms on a large-scale data set is *the curse of dimensionality* [8], since multimedia data is usually represented by features vectors with tens of thousands of dimensions. The volume and dimensions of large-scale data sets have led to challenges in both space and time. In order to achieve sublinear time and space algorithms for NNS, Approximate Nearest Neighbors (ANN) approaches are proposed, which sacrifice a small fraction of effectiveness to achieve higher efficiency, such as logarithmic ($\mathcal{O}(\log N)$), or even constant ($\mathcal{O}(1)$) query time. One popular family is the tree-based indexing approaches, which include KD tree [9], ball tree [10], metric tree [11], and vantage point tree [12]. However, there are also drawbacks of these tree-based approaches. One is the high storage requirement, and the other is the inefficiency in handling high-dimensional data.

To this end, mapping high-dimensional image descriptors to compact binary codes has been suggested, leading to considerable efficiency gains in both storage and similarity computation of images. The reason is simple: compact binary codes are much more efficient to store than floating-point feature vectors. Meanwhile, similarity based on Hamming distances between binary bits is much easier to compute than euclidean distances between real-valued features.

The benefits of binary encoding, also known as *Hashing* and *Quantization*, have motivated a tremendous amount of research in binary code generation such as [13], [14], [15], [16], [17] [18], [19], [20], [21], [4], [22], [23], [24], [25], [6], [26], [27], [28], [29], [30], [31], [32], [33]. A thorough survey is beyond the scope of this paper, interested readers could refer to [27]. Common in many methods, the first step of binary encoding leverages a linear mapping to project original features in high dimensions to lower dimensions. The representatives include Locality Sensitive Hashing (LSH) [13], Spectral Hashing (SH) [18], PCA Quantization (PCAQ) [4], Iterative Quantization (ITQ) [20], and Isotropic Hashing (IsoH) [34]. LSH uses random projections to form linear mapping, which is categorized into *data-independent* approaches since the used coding (hash) functions are fully independent of training data. Although learning-free, LSH requires long codes to achieve satisfactory accuracy. In contrast, *data-dependent* approaches can obtain high-quality compact codes by learning from training data. Specifically, PCAQ applies PCA to project input data onto a low-dimensional subspace, and then simply thresholds the projected data to generate binary bits each of which corresponds to a single PCA projection. Following PCAQ, SH, ITQ, and IsoH, all employ PCA to acquire low-dimensional data embedding, and then propose different post-processing schemes to produce binary bits. A common drawback of the above learning-driven binary encoding methods is the expensive computational cost of matrix manipulations.

In this paper, we demonstrate that the most time-consuming matrix operations encountered in code learning, typically data projection and rotation, can be performed in a more efficient manner. To this end, we propose a fast matrix multiplication algorithm using a sub-selection [35] technique to accelerate the learning of coding functions. Our algorithm is motivated by the observation that the degree of algorithm parameters is usually very small compared to the number of entire data samples. Therefore, we are able to determine these parameters by merely using partial data samples. This result is closely related to the matrix sketching for computation [36].

The contributions of this paper are fourfold: **(1)** To handle large-scale data, we propose a sub-selection based matrix multiplication algorithm and demonstrate its benefits theoretically. **(2)** We develop two fast quantization methods PCAQ-SS and ITQ-SS by combining the sub-selective algorithm with PCAQ and ITQ. **(3)** We also extend our approaches from linear embedding to non-linear kernel cases with two new approaches. **(4)** Extensive experiments are conducted to validate the efficiency and effectiveness of the proposed sub-selective quantization approaches, which indicate that these approaches can achieve acceleration up to 30 times in binary code learning yet with an imperceptible loss of accuracy.

This paper is an extended version of the work initially published in AAAI 2014 [37]. This paper differs greatly from the conference paper with additions that include: **1)** Improved methods: Feature embedding has been extended from linear to nonlinear using a kernel trick with two new kernelized binary encoding

• *The authors are with the University of Texas at Arlington, Arlington, TX, 76019 and Tencent AI Lab, Shenzhen 518057 China.*
E-mail: yeqing.li@mavs.uta.edu, wliu@ee.columbia.edu, jzhuang@uta.edu.

Manuscript received 9 May 2016; revised 27 Feb. 2017; accepted 30 Apr. 2017. Date of publication 30 May 2017; date of current version 14 May 2018.
(Corresponding author: J. Huang.)
Recommended for acceptance by J. Ye.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2710186

0162-8828 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

approaches proposed. **2)** More extensive experiments: New studies have been developed to the application of the sub-selection technique on the kernelized binary encoding approaches and we have included more experimental results to better demonstrate the computational efficiency and scalability of our method. **3)** Clearer explanation: The proposed sub-selective quantization approaches and related techniques are further explained in more details.

2 Background and Related Work

Before describing our methods, we will briefly introduce the binary code learning problem and two popular approaches.

Binary Encoding. Is trying to seek a coding function which maps a feature vector to short binary bits. Let $X \in \mathbb{R}^{n\times d}$ be the matrix of input data samples, and the ith data sample $x_i \in \mathbb{R}^{1\times d}$ be the ith row in X. In addition, X is made to be zero-centered. The goal is then to learn a binary code matrix $B \in \{-1,1\}^{n\times c}$, where c denotes the code length. The coding functions of several hashing and quantization methods can be formulated into $h_k(x) = sgn(xp_k)$ $(k = 1,\ldots,c)$, where $p_k \in \mathbb{R}^d$ and the sign function $sgn(\cdot)$ is defined as: $sgn(v) = 1$ if $v > 0$, $sgn(v) = -1$ otherwise. Hence, the coding process can be written as $B = sgn(XP)$, where $P = [p_1,\cdots,p_c] \in R^{d\times c}$ is the projection matrix.

PCA Quantization. [4] finds a linear transformation $P = W$ that maximizes the variance of each bit and makes the c bits mutually uncorrelated. W is obtained by running a Principal Components Analysis (PCA). Let $[W,\Lambda] = eig(\cdot,c)$ be a function which returns the first c eigenvalues in a diagonal matrix $S \in \mathbb{R}^{c\times c}$ and the corresponding eigenvectors as columns of $W \in \mathbb{R}^{d\times c}$. The whole procedure is summarized in Algorithm 1. While it is not a good coding method, its PCA step has been widely used as an initial step of many sophisticated coding methods. However, the computation of PCA involves multiplication with the high-dimensional matrix X, which consumes a considerable amount of memory and computation time. We will address the efficiency issue of PCAQ in next section.

Algorithm 1. PCA Quantization (PCAQ)

Input: Zero-centered data $X \in \mathbb{R}^{n\times d}$, code length c.
Output: $B \in \{-1,1\}^{n\times c}$, $W \in \mathbb{R}^{d\times c}$.
$cov = X^T X$;
$[W,\Lambda] = eig(cov,c)$;
$B = sgn(XW)$.

Iterative Quantization. [20] improves the quality of PCAQ by iteratively finding the optimal rotation matrix R on the projected data to minimize the quantization error. This is done through finding an appropriate orthogonal rotation by minimizing

$$Q(B,R) = \|B - VR\|_F^2, \tag{1}$$

where $V = XW$ is the PCA projected data. This equation is minimized using spectral clustering like the iterative quantization procedure [38]. The whole procedure is summarized in Algorithm 2, where $svd(\cdot)$ indicates a singular value decomposition. The ITQ method converges in a small number of iterations and is able to achieve high-quality binary codes compared with state-of-the-art coding methods. However, it involves not only multiplication with high-dimensional matrices (e.g., X^TX and B^TV) in the PCA step, but also those inside each quantization iteration, which makes it very slow in training. In next section, we will propose a method to overcome this drawback while preserving almost the same level of coding quality.

3 Methodology

According to our previous discussion, the common bottleneck of many existing methods is high dimensional matrix multiplication. However, dimensions of the product of these multiplication is relatively small. This motivated us to search for good approximation of those products using a subset of data, which resulted in our sub-selective matrix multiplication approach.

Algorithm 2. Iterative Quantization (ITQ)

Input: Zero-centered data $X \in \mathbb{R}^{n\times d}$, code length c, iteration number N.
Output: $B \in \{-1,1\}^{n\times c}$, $W \in \mathbb{R}^{d\times c}$.
$cov = X^T X$;
$[W,\Lambda] = eig(cov,c)$;
$V = XW$;
initialize R as an Orthogonal Gaussian Random matrix;
for $k = 1$ **to** N **do**
  $B = sgn(VR)$;
  $[S,\Lambda,\hat{S}] = svd(B^TV)$;
  $R = \hat{S}S^T$;
end for
$B = sgn(VR)$.

3.1 Sub-Selective Matrix Multiplication

The motivation behind sub-selective multiplication can be explained intuitively using data distribution. First of all, the rank r of the data matrix X is much smaller than n when $d \ll n$. Hence, all samples can be linearly represented by a small subset of all. In previous discussion, the quantization algorithms were set up to learn the parameters, i.e., W and R, that can transform data distribution according to specific criteria (e.g., variances). If data have close to uniform distribution, then a sufficient random subset can represent the full set well enough. Therefore we can find those parameters by solving the optimization problems in the selected subsets.

We begin with introduction to the notations of sub-selection. Let $\Omega \subset \{1,\ldots,n\}$ denote the indexes of selected rows of matrix ordered lexicographically and $|\Omega| = m$ denote the cardinality of Ω. With the same notations as Section 2, the sub-selection operation on X can be expressed as $X_\Omega \in \mathbb{R}^{m\times d}$ that consists of the row subset of X. For easy understanding we can consider X_Ω as $I_\Omega X$ where X is multiplied by a matrix $I_\Omega \in \{0,1\}^{m\times n}$ that consists of random row subset of the identify matrix I_n.

With sub-selection operation for matrix $Y \in \mathbb{R}^{n\times d_1}$ and $Z \in \mathbb{R}^{n\times d_2}$ where $d_1, d_2 \ll n$, sub-selective multiplication uses $\frac{n}{m}Y_\Omega^T Z_\Omega$ to approximate Y^TZ. Moreover, for a special case Y^TY, its sub-selection approximation is $\frac{n}{m}Y_\Omega^T Y_\Omega$. The complexity of multiplication is now reduced from $\mathcal{O}(nd_1d_2)$ to $\mathcal{O}(md_1d_2)$. Before we apply this method to binary quantization, we will first check to see if it's theoretically sound.

We will prove an error bound for sub-selective multiplication. Before providing our analysis, we first introduce a key result (Lemma 3.1 below) that will be crucial later analysis.

Lemma 3.1 (McDiarmid's Inequality [39]). *Let $X_1,\ldots,X_n$ be independent random variables, and assume f is a function for which there exists t_i, $i = 1,\ldots,n$ satisfying*

$$\sup_{x_1,\ldots,x_n,\hat{x}_i} |f(x_1,\ldots,x_n) - f(x_1,\ldots,\hat{x}_i,\ldots,x_n)| \le t_i, \tag{2}$$

where $\hat{x}_i$ indicates replacing the sample value x_i with any other of its possible values. Call $f(X_1,\ldots,X_n) := Y$. Then for any $\epsilon > 0$,

$$\mathbb{P}[Y \ge \mathbb{E}[Y] + \epsilon] \le \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n t_i^2}\right) \tag{3}$$

$$\mathbb{P}[Y \le \mathbb{E}[Y] - \epsilon] \le exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n t_i^2}\right). \tag{4}$$

Lemma 3.2 (Noncommutative Matrix Bernstein Inequality [40], [41] version 2). *Let $X_1, \ldots, X_m$ be independent zero-mean random matrices of dimension $d_1 \times d_2$. Suppose $\gamma = max\{\|\mathbb{E}[\sum_{k=1}^m X_k X_k^T]\|_2, \|\mathbb{E}[\sum_{k=1}^m X_k^T X_k]\|_2\}$ and $\|X_k\| \leq M$ almost surely for all k, where $\|\cdot\|_2$ is the matrix ℓ_2-norm (a.k.a. spectral norm). Then for any $\tau > 0$,*

$$\mathbb{P}\left[\|\sum_{k=1}^m X_k\|_2 > \tau\right] \leq (d_1+d_2)\exp\left(\frac{-\tau^2/2}{\gamma + M\tau/3}\right) \tag{5}$$

Let U be an $n \times r$ matrix whose columns span the r-dimensional subspace S. Let $P_S = U(U^TU)^{-1}U^T$ denotes the projection operator onto S. The "*coherence*" [42] of U is defined as

$$\mu(S) := \frac{n}{r}\max_j \|P_S e_j\|_2^2, \tag{6}$$

where e_j represents a standard basis element. $\mu(S)$ measures the maximum magnitude attainable by projecting a standard basis element onto S. Note that $1 \leq \mu(S) \leq \frac{n}{r}$. Let $z = [\|U_1\|_2, \ldots, \|U_i\|_2, \ldots, \|U_n\|_2]^T \in \mathbb{R}^n$, where each element of z is l_2-norm of one row in U. Thus, based on "*coherence*", we define "*row coherence*" to be the quantity

$$\phi(S) := \mu(z). \tag{7}$$

By plugging in the definition, we have $\phi(S) = \frac{n\|U\|_{2,\infty}^2}{\|U\|_F^2}$, where $\|\cdot\|_{2,\infty}$ means first compute the l_2-norm of each row then compute l_∞-norm of the result vector.

The key contribution of this paper is the following two theorems that form the analysis of bounds to sub-selective matrix multiplication. We start from the special case $Y_\Omega^T Y_\Omega$.

Theorem 3.1. *Suppose $\delta > 0$, $Y \in \mathbb{R}^{n\times d}$ and $|\Omega| = m$, then*

$$(1-\alpha_1)\frac{m}{n}\|Y\|_F^2 \leq \|Y_\Omega\|_F^2 \leq (1+\alpha_1)\frac{m}{n}\|Y\|_F^2, \tag{8}$$

with probability at least $1-2\delta$, where $\alpha_1 = \sqrt{\frac{2\phi_1(Y)^2}{m}\log(\frac{1}{\delta})}$ and $\phi_1(Y) = \frac{n\|Y\|_{2,\infty}^2}{\|Y\|_F^2}$.

Proof. We use McDiarmid's inequality from Lemma 3.1 for the function $f(X_1, \ldots, X_m) = \sum_{i=1}^m X_i$ to prove this. Set $X_i = \sum_{j=1}^d |Y_{\Omega(i),j}|^2$. Since $\sum_{j=1}^d |Y_{\Omega(i),j}|^2 \leq \|Y\|_{2,\infty}^2$ for all i, we have

$$\left|\sum_{i=1}^m X_i - \sum_{i\neq k} X_i - \hat{X}_k\right| = \left|X_k - \hat{X}_k\right| \leq 2\|Y\|_{2,\infty}^2. \tag{9}$$

We first calculate $\mathbb{E}[\sum_{i=1}^m X_i]$ as follows. Define $\mathbb{I}_{\{\}}$ to be the indicator function, and assume that the samples are taken uniformly with replacement

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^m X_i\right] &= \mathbb{E}\left[\sum_{i=1}^m\sum_{j=1}^d |Y_{\Omega(i,j)}|^2\right] \\ &= \sum_{i=1}^m\left[\mathbb{E}\left[\sum_{k=1}^n\sum_{j=1}^d |Y_{k,j}|^2 \mathbb{I}_{\{\Omega(i)=k\}}\right]\right] = \frac{m}{n}\|Y\|_F^2.\end{aligned} \tag{10}$$

Invoking the Lemma 3.1, the left hand side is

$$\mathbb{P}\left[\sum_{i=1}^m X_i \leq \mathbb{E}\left[\sum_{i=1}^m X_i\right] - \epsilon\right] = \mathbb{P}\left[\sum_{i=1}^m X_i \leq \frac{m}{n}\|Y\|_F^2 - \epsilon\right]. \tag{11}$$

We can let $\epsilon = \alpha_1 \frac{m}{n}\|y\|_F^2$ and then this probability is bounded by

$$\exp\left(\frac{-2\alpha_1^2(\frac{m}{n})^2\|Y\|_F^4}{4m\|Y\|_{2,\infty}^4}\right). \tag{12}$$

Thus, the resulting probability bound is

$$\mathbb{P}\left[\|Y_\Omega\|_F^2 \geq (1-\alpha_1)\frac{m}{n}\|Y\|_F^2\right] \geq 1 - \exp\left(\frac{-\alpha_1^2 m\|Y\|_F^4}{2n^2\|Y\|_{2,\infty}^4}\right). \tag{13}$$

Substituting our definitions of $\phi_1(Y) = \frac{n\|Y\|_{2,\infty}^2}{\|Y\|_F^2}$ and $\alpha_1 = \sqrt{\frac{2\phi_1(Y)^2}{m}\log(\frac{1}{\delta})}$ shows that the lower bound holds with probability at least $1-\delta$. The argument for the upper bound can be proved similarly. The theorem now follows by applying the union bound. □

Now we analyze the property of general case $Y_\Omega^T Z_\Omega$.

Theorem 3.2. *: Suppose $\delta > 0$, $Y \in \mathbb{R}^{n\times d_1}$, $Z \in \mathbb{R}^{n\times d_2}$ and $|\Omega| = m$, then*

$$\|Y_\Omega^T Z_\Omega - \frac{m}{n}Y^TZ\|_2 \leq \beta\frac{m}{n}\|Y^TZ\|_2, \tag{14}$$

with probability at least $1-\delta$, where $M = 2\sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2}$, $\gamma = \frac{m}{4n}M^2$ and $\delta = (d_1+d_2)\exp(\frac{-\beta\frac{m^2}{n^2}\|Y^TZ\|_2^2/2}{\frac{m}{4n}M^2 + M\beta\frac{m}{n}\|Y^TZ\|_2/3})$.

Proof. This theorem can be proved by involving Noncommutative Matrix Bernstein inequality. Let $S_i = \mathbb{E}[Y_{\Omega(i)}^T Z_{\Omega(i)}] \in \mathbb{R}^{d_1\times d_2}$ for $i \in \{1, \ldots, m\}$, where $\Omega(i)$ denotes the i^{th} sample index, $Y_{\Omega(i)} \in \mathbb{R}^{d_1\times 1}$ and $Z_{\Omega(i)} \in \mathbb{R}^{d_2\times 1}$. First, we have $S_i = \mathbb{E}\left[Y_{\Omega(i)}^T Z_{\Omega(i)}\right] = \mathbb{E}\left[\sum_{j=1}^n Y_j Z_j \mathbb{I}_{\{\Omega(i)=j\}}\right] = \sum_{j=1}^n Y_j Z_j \frac{1}{n} = \frac{1}{n}Y^TZ$.. That means S_i is actually independent of index i. Therefore, later on we will use $S = \mathbb{E}[Y_{\Omega(i)}^T Z_{\Omega(i)}] = \frac{1}{n}Y^TZ$ to denote these expectations. Let $X_i = Y_{\Omega(i)}^T Z_{\Omega(i)} - S$. Obviously, we have $\mathbb{E}[X_i] = 0$.

Then, we compute the upper bound of $\|X_i\|_2$

$$\begin{aligned}\|X_i\|_2 = \|Y_{\Omega(i)}^T Z_{\Omega(i)} - S\|_2 &\leq \|Y_{\Omega(i)}^T Z_{\Omega(i)}\|_2 + \|S\|_2 \\ &\leq 2\sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2} = M.\end{aligned} \tag{15}$$

Next, we bound $\|Y_{\Omega(i)}^T Z_{\Omega(i)}\|_2$ for all i. Observe that $\|Y_{\Omega(i)}\|_F = \|Y^T e_i\|_2 = \|P_{S_Y} e_i\|_2 \leq \sqrt{d_1\mu(S_Y)/n}$ by assumption, where S_Y refers to the subspace span by Y. Likewise, we have $\|Z_{\Omega(i)}\|_F \leq \sqrt{d_2\mu(S_Z)/n}$, where S_Z refers to the subspace span by Z. Thus,

$$\begin{aligned}\|Y_{\Omega(i)}^T Z_{\Omega(i)}\|_2 &\leq \|Y_{\Omega(i)}^T Z_{\Omega(i)}\|_F \leq \|Y_{\Omega(i)}\|_F \|Z_{\Omega(i)}\|_F \\ &\leq \sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2}.\end{aligned} \tag{16}$$

Now we compute the bound of $\|\mathbb{E}[\sum_{k=1}^m X_k X_k^T]\|_2$

$$\begin{aligned}&\|\mathbb{E}[\sum_{k=1}^m X_k X_k^T]\|_2 \\ &= \|\mathbb{E}[\sum_{k=1}^m (Y_{\Omega(i)}^T Z_{\Omega(i)} - S)(Y_{\Omega(i)}^T Z_{\Omega(i)} - S)^T]\|_2 \\ &= \|\sum_{k=1}^m \mathbb{E}[Y_{\Omega(i)}^T Z_{\Omega(i)} Z_{\Omega(i)}^T Y_{\Omega(i)}] - SS^T\|_2 \\ &\leq \max\left\{\|\sum_{k=1}^m \mathbb{E}[Y_{\Omega(i)}^T Z_{\Omega(i)} Z_{\Omega(i)}^T Y_{\Omega(i)}]\|_2, m\|SS^T\|_2\right\} \\ &\leq \frac{m}{n}d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2 = \frac{m}{4n}M^2.\end{aligned} \tag{17}$$

Therefore, let $M = 2\sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2}$, $\gamma = \frac{m}{4n}M^2$ and $\tau = \beta\frac{m}{n}\|Y^TZ\|_2$. Applying the Noncommutative Matrix

Bernstein Inequality, we have

$$\mathbb{P}\Big[\|Y_\Omega^T Z_\Omega - \frac{m}{n} Y^T Z\|_2 \leq \beta \frac{m}{n} \|Y^T Z\|_2\Big] \leq 1 - \delta, \tag{18}$$

where $\delta = (d_1 + d_2)\exp\left(\frac{-\beta m^2 \|Y^T Z\|_2^2/2}{\frac{m}{4n}M^2 + M\beta\frac{m}{n}\|Y^T Z\|_2/3}\right)$. □

The above two theorems prove that the product of sub-selective multiplication will be very close the original product of full data with high probability.

3.2 Case Studies

3.2.1 Unsupervised Sub-Selective Quantization

With the theoretical guarantee, we are now ready to apply sub-selective multiplication on existing quantization methods, i.e., PCAQ [4] and ITQ [20]. A common initial step for each of them is PCA projection (e.g., Algorithms 1 and 2). The time complexity for matrix multiplication $X^T X$ is $\mathcal{O}(nd^2))$ when $d < n$. For large n, this step could take up a considerable amount of time. Hence, we can approximate it by $\frac{1}{m} X_\Omega^T X_\Omega$, which is surprisingly the covariance matrix of the selected samples. From a statistics point of view, this could be intuitively interpreted as using the variance matrix of a random subset of samples to approximate the covariance matrix of full ones when the data is redundant. Now the time complexity is only $\mathcal{O}(md^2)$, where $m \ll n$ in a large data set. For ITQ, the learning process includes dozens of iterations to find rotation matrix R (Algorithm 2 line 7 to 11). We approximate R with $\hat{R} = S_r S_l$, where $S_l \Lambda S_r = B_\Omega^T V_\Omega$ is the SVD of $B_\Omega^T V_\Omega$: B_Ω and V_Ω are a sub-selection version of B and V in Algorithm 2 respectively. The time complexity of compute R is reduced from $\mathcal{O}(nc^2)$ to $\mathcal{O}(mc^2)$.

By replacing corresponding steps in original methods, we get two sub-selective quantization methods corresponding to PCAQ and ITQ, which are named PCAQ-SS, ITQ-SS. ITQ-SS is summarized in Algorithm 3. PCAQ-SS is the same as first 5 lines in Algorithm 3 plus one encoding step $B = sgn(V)$. It is omitted because of the page limits. Complexity of original ITQ is $\mathcal{O}(nd^2 + (p+1)nc^2)$. In contrast, complexity of ITQ-SS is reduced to $\mathcal{O}(md^2 + pmc^2 + nc^2)$. The acceleration can be seen more clearly in the experimental results in the next section.

Algorithm 3. ITQ with Sub-Selection (ITQ-SS)

Input: Zero-centered data $X \in \mathbb{R}^{n \times d}$, code length c, iteration number p.
Output: $B \in \{-1, 1\}^{n \times c}$, $W \in \mathbb{R}^{d \times c}$.

1) Uniformly randomly generate $\Omega \subset [1:n]$;
2) $X_\Omega = \Omega \odot X$;
3) $cov = X_\Omega^T X_\Omega$;
4) $[W, \Lambda] = eig(cov, c)$;
5) $V = XW$;
6) Initialize R as an orthogonal Gaussian random matrix;

for $k = 1$ **to** p **do**
  uniformly randomly generate $\Omega \subset [1:n]$;
  compute V_Ω;
  $B_\Omega = sgn(V_\Omega R)$;
  $[S, \Lambda, \hat{S}] = svd(B_\Omega^T V_\Omega)$;
  $R = \hat{S} S^T$;
end for
7. $B = sgn(VR)$.

3.2.2 Kernelized Sub-Selective Quantization

The previous discussions are limited to linear embedding of the data. Here we will extend our case studies to nonlinear cases. The Kernel trick [43] is usually used for mapping data point x_i to a higher or even infinity dimension $\phi(x_i)$. Then the kernel PCA (KPCA) [43] is used for nonlinear embedding by performing the eigendecomposition on the kernel matrix. The kernel matrix K is an $n \times n$ matrix, where each element k_{ij} is the inner product of the feature $k_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. Here, we take the Guassian kernel $K(x, y) = \exp(-\|x - y\|^2)/(2\sigma^2)$ for example. Recently, the explicit approximation of the Gaussian kernel has attracted much attentions. Therefore, we employ one of the most popular approximations referred to as *Random Fourier Feature* (RFF) [44]. Based on the definition of RFF, the feature vector of each data point x is computed as

$$\Phi^D(x) = [\Phi_{w_1,b_1}(x), \Phi_{w_2,b_2}(x), \ldots, \Phi_{w_D,b_D}(x)], \tag{19}$$

where D is the user-specific dimension of RFF and $\Phi_P w_1, d_1(x)$ is the value of one single coordinate of the feature vector and $w_i, b_i \forall i \in [1, D]$ represents the projected parameters. $\Phi_P w, d(x)$ is defined as

$$\Phi_P w, d(x) = \sqrt{2} \cos xw + b, \tag{20}$$

where w is a random projection vector drawn from normal distribution $\mathcal{N}(0, \frac{1}{2\sigma^2} I)$, and b is another random vector drawn from uniform distribution $\mathcal{U}[0, 2\pi]$. Note that explicit approximation for many other types of kernels have been used [45], [46], [47]; however, we only focus on the RFF for approximation of the Gaussian kernel here.

After transforming the original features to RFF, we can perform KPCA on the RFF, which is similar to the ordinary PCA. By combing the KPCA with ITQ, one can get a kernelized quantization approach KPCA-ITQ [20]. It is claimed that this nonlinear embedding approach can improve performance. Another benefit of using the kernel embedding is that one can achieve bit length greater than the original feature dimension due to the higher dimensional embedding. However, the kernel approach also leads to higher computational complexity due to the construction of the kernel and the embedding on higher dimensional features. Therefore, if we can apply the proposed sub-selective technique on the KPCA as well as the ITQ process, we can obtain a more efficient kernelized quantization approach, dubbed KPCA-ITQ-SS.

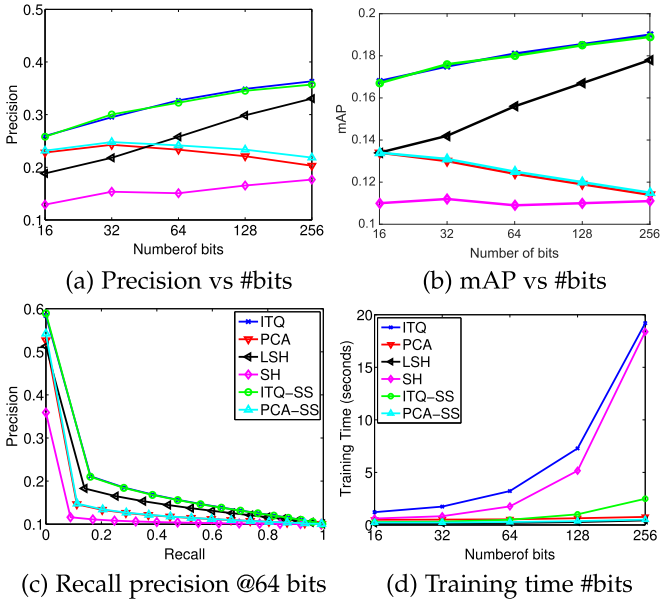
4 Evaluations

4.1 Experimental Settings

In this section, we evaluate the sub-selective quantization approaches on three public data sets: *CIFAR* [48], [1]*MNIST*[2] and *Tiny-1M* [4]. *CIFAR* consists of 60K 32×32 color images that have been manually labeled to ten categories. Each category contains 6K samples. Each image in CIFAR is assigned to one mutually exclusive class label and represented by a 512-dimension GIST feature vector [49]. *MNIST* consists of 70K samples of a 784-dimension feature vector associated with digits from '0' to '9'. The true neighbors are defined as semantic neighbors based on the associated digit labels. *Tiny-1M* consists of one million images. Each image is represented by a 384-dimension GIST vector. Since manually labels are not available on Tiny-1M, euclidean neighbors are computed and used as ground truth for nearest neighbor search.

We compare the proposed methods, *PCAQ-SS* and *ITQ-SS*, with their corresponding unaccelerated methods, *PCAQ* [4] and *ITQ* [20]. We also compare our methods to two baseline methods that follow a similar quantization scheme $B = sgn(X\tilde{W})$: **1)** *LSH* [13], where $\tilde{W}$ is a Gaussian random matrix and **2)** *SH* [18], which is based on quantizing the values of analytical eigenfunctions computed along PCA directions of the data. All the compared codes are provided by the authors.

1. http://www.cs.toronto.edu/ kriz/cifar.html
2. http://yann.lecun.com/exdb/mnist/



(a) Precision vs #bits (b) mAP vs #bits

(c) Recall precision @64 bits (d) Training time #bits

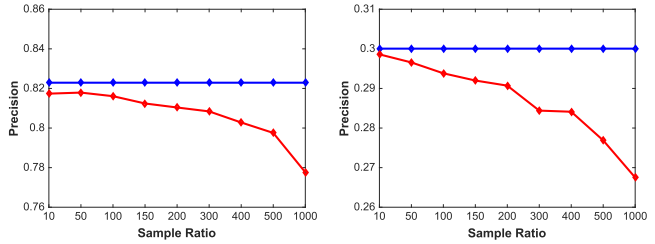
Fig. 1. The results on CIFAR. All the subfigures share the same legends.

Two types of evaluation are conducted following [20]. First, semantic consistency of codes is evaluated for different methods while class labels are used as ground truth. We reported four measures, the *average precision of top 100 ranked images* for each query, *mean average precision*, *recall-precision curve* and *training time*, in CIFAR and MNIST. Second, we used the generated codes for nearest neighbor search, where euclidean neighbors are used as ground truth. This experiment is conducted on a Tiny-1M data set. We reported three measures: *average precision of top 5 percent ranked images* for each query and *training time*. For both types of evaluation, the query algorithm and corresponding structure of binary code were the same, so *testing time* was exactly the same for all the methods except SH. Hence, it's omitted from the results. All our experiments were conducted on a desktop computer with a 3.4 GHz Intel Core i7 and 12 GB RAM.

4.2 Unsupervised Binary Encoding Results on CIFAR

The CIFAR data set is partitioned into two parts: 59K images as a training set and 1K images as a test query set evenly sampled from ten classes. We uniformly and randomly generated our sub-selective matrix Ω with cardinality equal to $1/40$ of number of the data points, i.e., $|\Omega| = m = n/40$.

Figs. 1a and 1b show complete precision of top 100 ranked images and the mean average precision (mAP) over 1K query images for different numbers of bits. Fig. 1c shows recall-precision curve of 64 bits. For these three metrics, ITQ and ITQ-SS have the best performance. Both sub-selective methods (PCAQ-SS and ITQ-SS) preserve the performance of original methods (i.e., PCAQ and ITQ). Our results indicate that sub-selection preserves semantic consistency of the original coding method. Fig. 1d shows the training time of the two methods. Our method is about *4 to 8 times faster* than ITQ [20]. Original ITQ is the slowest among all the comparing methods, while the speed of the accelerated version ITQ-SS is comparable, if not superior, to the fastest methods. This is due to ITQ-SS reducing the dimension of the problem from a function of n to that of m, where $m \ll n$. Figs. 2a and 2b show precision comparison between ITQ and ITQ-SS when changing the sampling ratio. We can see that the precisions drop slowly when decreasing the number of samples. These results validate the benefits of sub-selection to preserve the performance of the original method with far less training cost.



(a) Precision @32 bits vs sample ratio@MNIST. (b) Precision @32 bits vs sample ratio@CIFAR.

Fig. 2. The results on MNIST and CIFAR at 32 bits. Comparison of the deviation of ITQ-SS and ITQ when changing sampling ratio.

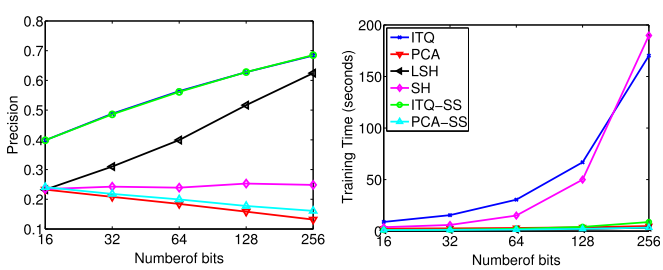
4.3 Unsupervised Binary Encoding Results on Tiny-1M

For experiment without labeled ground truth, a separate subset of 2K images with 80 million images are used as the test set while another one million images are used as the training set. We uniformly and randomly generate our sub-selective matrix Ω with cardinality equal to $1/1000$ of number of data points, i.e., $|\Omega| = m = n/1000$. Fig. 3a shows complete precision of the top 5 percent ranked images and mean average precision (mAP) over 1K query images for different numbers of bits. The difference between sub-selective methods (i.e., PCAQ-SS, ITQ-SS) and their counterparts (i.e., PCAQ, ITQ) were less than 1 percent. Fig. 3b shows the training time of the two methods. ITQ-SS achieved an even bigger speed advantage, which is about *10 to 30 times faster* than ITQ. This is because the larger data set samples were more redundant, making it possible to use smaller portions of data.

4.4 Results of Kernel Embedding Binary Encoding

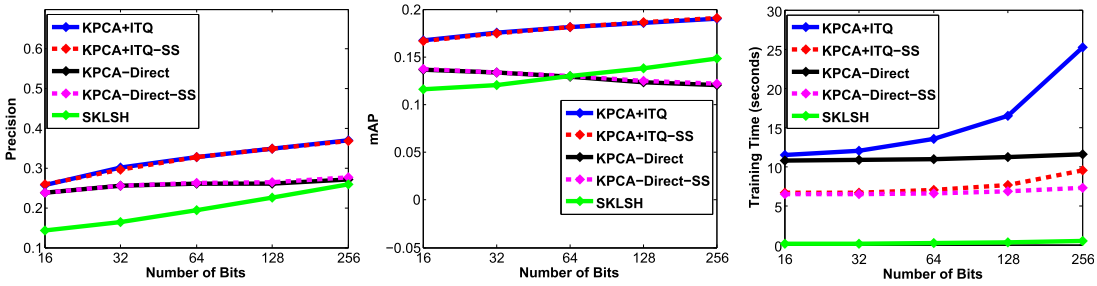
In this section, we evaluate the sub-selecitve version of our kernelized quantization approach KPCA+ITQ-SS and compared it with the baseline approaches. Similar to the linear embedding experiments, we also evaluated the KPCA-Direst-SS that is the sub-selective version of KPCA-Direst quantization. KPCA-direct is similar to PCAQ [20] except that instead of performing PCA, it uses KPCA. We also added another kernelized quantization approach SKLSH [16] as a baseline. We conducted experiments on the MNIST and the CIFAR data set and evaluated the precision, mAP and training time. The results of quantization code, with a bit length that varied from 32 to 256, are reported. For the RFF feature, we set the mapping dimension $D = 3000$ for the RFF feature in all experiments. One exception of the mapping dimension setting is SKLSH [16], which by definition uses the bit length as the mapping dimension. Another important hyper-parameter is the bandwidth σ of the Gaussian kernel. We followed the setting in [20] that uses the mean distance of the 50th nearest neighbor. Although in our experience, we did discover that further setting, changing σ to a smaller value will bring about performance improvement; however, we did not further fine-tuning the parameter.

Figs. 4 and 5 showed experimental results on CIFAR and MNIST, respectively. Figs. 4b and 4a show that the subselective algorithms,



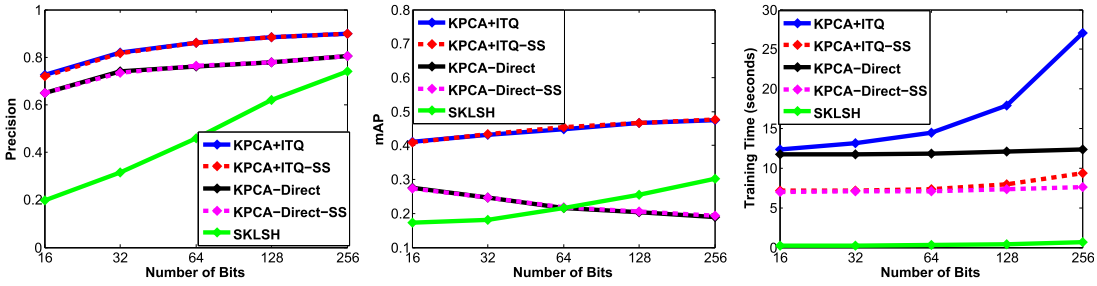
(a) Precision vs #bits. (b) Training time vs #bits.

Fig. 3. The results on Tiny-1M. All the subfigures share the same set of legends.



(a) Precision vs #bits. (b) mAP vs #bits. (c) Training time vs #bits.

Fig. 4. The results of kernel embedding binary encoding on CIFAR.



(a) Precision vs #bits. (b) mAP vs #bits. (c) Training time vs #bits.

Fig. 5. The results of kernel embedding binary encoding on MNIST.

KPCA+ITQ-SS and KPCA-Direct-SS, can achieve almost the same level of accuracy/precision as their non-subselecvie counterpart. And the algorithms that use ITQ or ITQ-SS have achieved the best performance among all, which is consistency with existing studies. Fig. 4c shows the running time. The figure clearly indicates that the sub-selective technique greatly reduced the computational cost of the training phrase. Similar trends can also be also witnessed in the results on MNIST (Figs. 5a, 5b, and 5c).

One might notice that in Figs. 4c and 5c), KPCA-ITQ-SS, KPCA-Direct-SS and KPCA-Direct are relatively flat on the plot. In other words, the computational cost grows slowly as the number of bits grow. This is due to the fact that a considerable amount of time is spent on the RFF generation. Specifically, it is the cost of computing feature projection and trigonometric function on the training set. After utilizing the sub-selective technique, the feature projection process becomes one of the bottlenecks. In practice, this can be easily parallelized by many off-the-shelf parallel software tools, e.g., Hadoop, Spark, GPGPU. However, we will omit an analysis this improvement since it is not the focus of this paper.

5 Discussion and Conclusion

All of the experimental results presented herein verified the benefits of the proposed sub-selective quantization technique whose parameters can be automatically learned from a subset of the original data set. The proposed PCAQ-SS and ITQ-SS methods can achieve almost the same quantization quality as PCAQ and ITQ with only a small portion of training time. The advantage in training time is more prominent on larger data sets, e.g., 10 to 30 times faster on Tiny-1M. Hence, for larger data sets good quantization quality can be achieved with an even lower sampling ratio. Furthermore, we extended our approach to a kernelized scenario approaches and proposed two novel binary encoding algorithms. Similar acceleration has been achieved in these algorithms compared to their original counterparts.

One may notice that the speed-up ratio is not as the same as the sampling ratio. This is because the training process of quantization includes not only finding the coding parameters but also generating the binary codes of the input data set. The latter inevitably involves operations upon the whole data set, which requires a considerable number of matrix multiplications. The proposed sub-selective quantization technique represents one single step requiring matrix multiplication, thus enabling an easy acceleration by using parallel or distributed computing techniques.

We accredit the success of the proposed sub-selective quantization technique to the effective use of sub-selection in accelerating the quantization optimization that involves large-scale matrix multiplications. Moreover, the benefits of sub-selection were theoretically demonstrated. As a case study of sub-selective quantization, we found that ITQ-SS can accomplish the same level of coding quality with significantly reduced training time in contrast to existing methods. The extensive image retrieval results on large image corpora scaling up to one million further empirically verified the speed gain of sub-selective quantization.

Acknowledgments

This work was partially supported by US National Science Foundation IIS-1423056, CMMI-1434401, CNS-1405985, IIS-1718853 and the NSF CAREER grant IIS-1553687.

References

[1] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[2] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[3] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2143–2157, Dec. 2009.

[4] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[5] S. Korman and S. Avidan, "Coherency sensitive hashing," in *Proc. IEEE Conf. Comput. Vis.*, 2011, pp. 1607–1614.

[6] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[7] G. Shakhnarovich, P. Indyk, and T. Darrell, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. Cambridge, MA, USA: MIT Press, 2006.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.

[9] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[10] S. M. Omohundro, *Efficient Algorithms with Neural Network Behavior*. Champaign, IL, USA: Dept. Comput. Sci., Univ. Illinois at Urbana-Champaign, 1987.

[11] J. K. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Inf. Process. Lett.*, vol. 40, no. 4, pp. 175–179, 1991.

[12] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *Proc. 4th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1993, vol. 93, no. 194, pp. 311–321.

[13] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 459–468.

[14] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[15] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization for approximate nearest neighbor search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2946–2953.

[16] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.

[17] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.

[18] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. 21st Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.

[19] Y. Weiss, R. Fergus, and A. Torralba, "Multidimensional spectral hashing," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 340–353.

[20] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[21] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[22] Y. Mu, J. Shen, and S. Yan, "Weakly-supervised hashing in kernel space," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3344–3351.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

[25] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 353–360.

[26] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2074–2081.

[27] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—a survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.

[28] M. Ou, P. Cui, J. Wang, F. Wang, and W. Zhu, "Probabilistic attributed hashing," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2894–2900.

[29] V. Erin Liong , J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2475–2483.

[30] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1556–1564.

[31] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3270–3278.

[32] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.

[33] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 37–45.

[34] W. Kong and W.-J. Li, "Isotropic hashing," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1646–1654.

[35] Y. Li, C. Chen, and J. Huang, "Transformation-invariant collaborative sub-representation," in *Proc. 22th Int. Conf. Pattern Recognit.*, 2014, pp. 3738–3743.

[36] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *Foundations and Trends® Theoretical Comput. Sci.*, vol. 10, no. 1–2, pp. 1–157, 2014.

[37] Y. Li, C. Chen, W. Liu, and J. Huang, "Subselective quantization for large-scale image search," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2803–2809.

[38] S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 313–319.

[39] C. McDiarmid , "On the method of bounded differences," *Surveys Combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[40] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing," *Phys. Rev. Lett.*, vol. 105, no. 15, 2010, Art. no. 150401.

[41] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learn. Res.*, vol. 12, pp. 3413–3430, 2011.

[42] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[43] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Artificial Neural Networks*. Berlin, Germany: Springer, 1997, pp. 583–588.

[44] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1177–1184.

[45] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[46] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2297–2304.

[47] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, Mar. 2012.

[48] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[49] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.