Sub-Selective Quantization for Large-Scale Image Search

Yeqing Li[1], Chen Chen[1], Wei Liu[2], Junzhou Huang[1]*
[1] University of Texas at Arlington, Arlington, TX, 76019, USA
[2]IBM T. J. Watson Research Center, 10598, NY, USA

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 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 two popular quantization techniques PCA Quantization (PCAQ) and Iterative Quantization (ITQ). 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.

Introduction

Similarity search has stood as a fundamental technique used in many vision related applications including object recognition (Torralba, Fergus, and Weiss 2008; Torralba, Fergus, and Freeman 2008), image retrieval (Kulis, Jain, and Grauman 2009)(Wang, Kumar, and Chang 2012), image matching (Korman and Avidan 2011)(Strecha et al. 2012), etc. The explosive growth of visual content on the Internet has made this task more challenging due to the high storage and computation overhead. 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, and meanwhile similarity based on Hamming distances among binary bits is much easier to compute than Euclidean distances among real-valued features.

The benefits of binary encoding, also known as *Hashing* and *Quantization* in literature, have motivated a tremendous amount of research in binary code generation such as (Andoni and Indyk 2006)(Jegou, Douze, and Schmid 2011)(Ge et al. 2013)(Raginsky and Lazebnik 2009)(Kulis and Grauman 2012) (Weiss, Torralba, and Fergus 2008)(Weiss, Fergus, and Torralba 2012)(Gong et al. 2013)(Liu et al. 2011) (Mu, Shen, and Yan 2010)(Wang, Kumar, and Chang 2012)(Hinton and Salakhutdinov 2006)(Kulis and Darrell 2009) (Norouzi and Blei 2011) (Strecha et al. 2012)(Liu et al. 2012). The main challenge of these approaches is how to effectively incorporate domain knowledge into traditional models (Huang, Huang, and Metaxas 2009), and how to efficiently solve them (Huang et al. 2011). The composite prior models are promising solutions because of their flexibility in modeling prior knowledge and their computational efficiency (Huang et al. 2011; Huang, Zhang, and Metaxas 2011). Common in many methods, the first step has been adopted to leverage a linear mapping to project original features in high dimensions to lower dimensions. The representatives include Locality Sensitive Hashing (LSH) (Andoni and Indyk 2006), Spectral Hashing (SH) (Weiss, Torralba, and Fergus 2008), PCA Quantization (PCAQ) (Wang, Kumar, and Chang 2012), Iterative Quantization (ITQ) (Gong et al. 2013), and Isotropic Hashing (IsoH) (Kong and Li 2012). LSH uses random projections to form such a 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 the input data onto a low-dimensional subspace, and 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 a low-dimensional data embedding, and then propose different postprocessing schemes to produce binary bits. A common drawback of the above learning-driven hashing methods is the expensive computational cost in 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 (Li, Chen,

[*] Corresponding author Email: jzhuang@uta.edu

and Huang 2014) technique to accelerate the learning of coding functions. Our algorithm is motivated by the observation that the degree of the algorithm parameters is usually very small compared to the number of entire data samples. Therefore, we are able to determine these parameters merely using partial data samples.

The contributions of this paper are three-folds: **(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)** Extensive experiments are conducted to validate the efficiency and effectiveness of the proposed PCAQ-SS and ITQ-SS, which indicate that ITQ-SS can achieve an up to 30 times acceleration of binary code learning yet with an imperceptible loss of accuracy.

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 i-th data sample $x_i \in \mathbb{R}^{1 \times d}$ be the i-th row in X. Additional, 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 $h_k(x) = sgn(xp_k)$ ($k = 1, \dots, c$), 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 (PCAQ) (Wang, Kumar, and Chang 2012) 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 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 widely used as an initial step of many sophisticated coding methods. However, the computation of PCA involves a multiplication with high-dimensional matrix X, which consumes considerable amount of memory and computation time. We will address the efficiency issue of PCAQ in the 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 (ITQ) (Gong et al. 2013) 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 the spectral-clustering like iterative quantization procedure (Yu and Shi 2003). The whole procedure is summarized in Algorithm 2, where $svd(\cdot)$ indicates 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 multiplications with high-dimensional matrices (*e.g.*, $X^T X$ and $B^T V$) in the PCA step, but also those inside each quantization iteration, which makes it very slow in training. In the next section, we will propose a method to overcome this drawback while preserving almost the same level of coding quality.

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
    $B = sgn(VR)$;
    $[S, \Lambda, \hat{S}] = svd(B^T V)$;
    $R = \hat{S} S^T$;
end for
$B = sgn(VR)$.
```

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 motivates us to search for good approximation of those products using a subset of data, which results in our sub-selective matrix multiplication approach.

Sub-selective Matrix Multiplication

The motivation behind sub-selective multiplication can be explain intuitively using data distribution. First of all, the data matrix X is low-rank compared to n when $d << n$. Hence, all samples can be linear represented by a small subset of all. In previous discussion, the quantization algorithms try to learn the parameters, i.e. W and R, that can transform data distribution according to specific criteria (e.g. variances). If data are distributed closely to uniform, 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 $\Omega \subset \{1, \dots, n\}$ denotes the indexes of selected rows of matrix ordered lexicographically and $|\Omega| =$

m denotes the cardinality of Ω. With the same notations as previous section, the sub-selection operation on X can be expressed as $X_\Omega \in \mathbb{R}^{m\times d}$ that consists of row subset of X. For easy understanding we can consider X_Ω as $I_\Omega X$ where X multiply 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 use $\frac{n}{m} Y_\Omega^T Z_\Omega$ to approximate $Y^T Z$. And for a special case $Y^T Y$, its sub-selection approximation is $\frac{n}{m} Y_\Omega^T Y_\Omega$. The complexity of multiplication is now reduced from $O(nd_1d_2)$ to $O(md_1d_2)$. Before we apply this methods to binary quantization, we will first examine if it's theoretically sound.

We will prove a bound for sub-selective multiplication. Before providing our analysis, we first introduce a key result (Lemma 1 below) that will be crucial for the later analysis.

Lemma 1. *(McDiarmid's Inequality (McDiarmid 1989)): Let $X_1, ..., X_n$ be independent random variables, and assume f is a function for which there exist t_i, $i = 1, ..., n$ satisfying*

$$\sup_{x_1,...,x_n,\hat{x}_i} |f(x_1, ..., x_n) - f(x_1, ..., \hat{x_i}, ..., x_n)| \le t_i \quad (2)$$

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

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

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

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*" (Candès and Recht 2009) of U is defined to be

$$\mu(S) := \frac{n}{r} \max_j \|P_S e_j\|_2^2, \quad (5)$$

where e_j represents a standard basis element. $\mu(S)$ measure the maximum magnitude attainable by projecting a standard basis element onto S. Note that $1 \le \mu(S) \le \frac{n}{r}$. Let $z = [\|U_1\|_2, \dots, \|U_i\|_2, \dots, \|U_n\|_2]^T \in \mathbb{R}^n$, 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

$$\phi(S) := \mu(z). \quad (6)$$

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

$$(1 - \alpha_1)\frac{m}{n}\|Y\|_F^2 \le \|Y_\Omega\|_F^2 \le (1 + \alpha_1)\frac{m}{n}\|Y\|_F^2 \quad (7)$$

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 1 for the function $f(X_1, \dots, X_m) = \sum_{i=1}^m X_i$ to prove this. Set $X_i = \sum_{j=1}^d |Y_{\Omega(i),j}|^2$. Let $\|\cdot\|_1$ denotes the l_1 norm of matrix. Since $\sum_{j=1}^d |Y_{\Omega(i),j}|^2 \le \|Y\|_{2,\infty}^2$ for all i, we have

$$\left|\sum_{i=1}^m X_i - \sum_{i\neq k} X_i - \hat{X}_k\right| = \left|X_k - \hat{X}_k\right| \le 2\|Y\|_{2,\infty}^2. \quad (8)$$

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.

$$\mathbb{E}\left[\sum_{i=1}^m X_i\right] = \mathbb{E}\left[\sum_{i=1}^m \sum_{j=1}^d |Y_{\Omega(i,j)}|^2\right]$$
$$= \sum_{i=1}^m \left[\mathbb{E}\left[\sum_{k=1}^n \sum_{j=1}^d |Y_{k,j}|^2 \mathbb{I}_{\{\Omega(i)=k\}}\right]\right] = \frac{m}{n}\|Y\|_F^2. \quad (9)$$

Invoking the Lemma 1, the left hand side is

$$\mathbb{P}\left[\sum_{i=1}^m X_i \le \mathbb{E}\left[\sum_{i=1}^m X_i\right] - \epsilon\right] = \mathbb{P}\left[\sum_{i=1}^m X_i \le \frac{m}{n}\|Y\|_F^2 - \epsilon\right]. \quad (10)$$

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

$$\exp\left(\frac{-2\alpha^2(\frac{m}{n})^2\|Y\|_F^4}{4m\|Y\|_{2,\infty}^4}\right) \quad (11)$$

Thus, the resulting probability bound is

$$\mathbb{P}\left[\|Y_\Omega\|_F^2 \ge (1-\alpha)\frac{m}{n}\|Y\|_F^2\right] \ge 1 - \exp\left(\frac{-\alpha^2 m\|Y\|_F^4}{2n^2\|Y\|_{2,\infty}^4}\right). \quad (12)$$

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 analysis the property of general case $Y_\Omega^T Z$.

Theorem 2. *: Suppose $\delta > 0$, $Y \in \mathbb{R}^{n\times d_1}$, $Z \in \mathbb{R}^{n\times d_2}$ and $|\Omega| = m$, then*

$$(1-\beta_1)^2(\frac{m}{n})^2\|Y^TZ\|_F^2 \le \|Y_\Omega^T Z_\Omega\|_F^2 \le (1+\beta_2)^2\frac{m}{n}\|Y^TZ\|_F^2 \quad (13)$$

with probability at least $1 - 2\delta$, where $\beta_1 = \sqrt{\frac{2nd_1d_2\mu(S_Y)\mu(S_Z)}{m^2\|Y^TZ\|_F^2}\log\left(\frac{1}{\delta}\right)}$ and $\beta_2 = \sqrt{\frac{2d_1d_2\mu(S_Y)\mu(S_Z)}{m\|Y^TZ\|_F^2}\log\left(\frac{1}{\delta}\right)}$.

Proof. This theorem can be proved by involving McDiarmid's inequality in similar fashion to the proof of Theorem 1. Let $X_i = Y_{\Omega(i)}^T Z_{\Omega(i)} \in \mathbb{R}^{d_1\times d_2}$, where $\Omega(i)$ denotes the i^{th} sample index, $Y_{\Omega(i)} \in \mathbb{R}^{d_1\times 1}$ and $Z_{\Omega(i)} \in \mathbb{R}^{d_2\times 1}$.

Let our function $f(X_1,\dots,X_m) = \|\sum_{i=1}^m X_i\|_F = \|Y_\Omega^T Z_\Omega\|_F$. First, we need to bound $\|X_i\|$ 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,

$$\begin{aligned}\|X_i\|_F = \|Y_{\Omega(i)}^T Z_{\Omega(i)}\|_F &\leq \|Y_{\Omega(i)}\|_F \|Z_{\Omega(i)}\|_F \\ &\leq \sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2}. \end{aligned} \tag{14}$$

Then $|f(X_1,\dots,X_m) - f(X_1,\dots,\hat{X}_K,\dots,X_m)|$ is bounded by

$$\begin{aligned}&\left|\left\|\sum_{i=1}^m X_i\right\|_F - \left\|\sum_{i\neq k} X_i + \hat{X}_k\right\|_F\right| \\ \leq& \|X_k - \hat{X}_k\|_F \leq \|X_k\|_F + \|\hat{X}_k\|_F \\ \leq& 2\sqrt{d_1 d_2 \mu(S_Y)\mu(S_Z)/n^2}, \end{aligned} \tag{15}$$

where the first two inequalities follow from the triangular inequality. Next we calculate the bound for $\mathbb{E}\left[f(X_1,\dots,X_m)\right] = \mathbb{E}[\|\sum_{i=1}^m X_i\|_F]$. Assume again that the samples are taken uniformly with replacement.

$$\mathbb{E}\left[\left\|\sum_{i=1}^m X_i\right\|_F^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^m Y_{\Omega(i)}^T Z_{\Omega(i)}\right\|_F^2\right]$$

$$= \sum_{k_1=1}^{d_1}\sum_{k_2=1}^{d_2} \mathbb{E}\left[\sum_{i=1}^m\sum_{j=1}^n Y_{k_1,j}^2 Z_{k_2,j}^2 \mathbb{I}_{\{\Omega(i)=j\}}\right] \tag{16}$$

$$= \sum_{k_1=1}^{d_1}\sum_{k_2=1}^{d_2} m \sum_{j=1}^n Y_{k_1,j}^2 Z_{k_2,j}^2 \frac{1}{n} = \frac{m}{n}\|Y^T Z\|_F^2 \tag{17}$$

The step (16) follows because of our assumption that sampling if uniform with replacement.

Since $\mathbb{E}[\|\sum_{i=1}^m X_i\|_F] \leq \mathbb{E}[\|\sum_{i=1}^m X_i\|_F^2]^{1/2}$ by Jensen's inequality, we have $\mathbb{E}\|\sum_{i=1}^m X_i\|_F] \leq \sqrt{\frac{m}{n}}\|Y^T Z\|_F$.

Using Jensen's inequality and indicator function in similar fashion, we also have bound for the left side:

$$\mathbb{E}\left[\|\sum_{i=1}^m X_i\|_F\right] \geq \left\|\sum_{i=1}^m \mathbb{E}\left[X_i\right]\right\|_F = \frac{m}{n}\|Y^T Z\|_F \tag{18}$$

Letting $\epsilon_1 = \beta_1 \frac{m}{n}\|Y^T Z\|_F$ and plugging into Equation (4), we then have that probability is bounded by $\exp\left(\frac{-2\beta_1^2(\frac{m}{n})^2\|Y^T Z\|_F^2}{4nd_1d_2\mu(S_Y)\mu(S_Z)/n^2}\right)$ Thus, the resulting probability bound is $\mathbb{P}\left[\|Y_\Omega^T Z_\Omega\|_F^2 \geq (1-\beta_2)^2(\frac{m}{n})^2\|Y^T Z\|_F^2\right]$ $\geq 1 - \exp\left(\frac{-\beta_1^2 m^2\|Y^T Z\|_F^2}{2nd_1d_2\mu(S_Y)\mu(S_Z)}\right)$ Substituting our definitions of $\mu(S_Y)$, $\mu(S_Z)$ and β_1 shows that the lower bound holds with probability at least $1-\delta$.

Letting $\epsilon_2 = \beta_2\sqrt{\frac{m}{n}}\|Y^T Z\|_F$ and with similar fashion we can obtain the probability of uppper bound: $\mathbb{P}\left[\|Y_\Omega^T Z_\Omega\|_F^2 \leq (1+\beta_2)^2\frac{m}{n}\|Y^T Z\|_F^2\right]$ $\geq 1 - \exp\left(\frac{-\beta_2^2 m\|Y^T Z\|_F^2}{2d_1d_2\mu(S_Y)\mu(S_Z)}\right)$

Substituting our definitions of $\mu(S_Y)$, $\mu(S_Z)$ and β_2 shows that the upper bound holds with probability at least $1-\delta$. The theorem now follows by applying the union bound, completing the proof. □

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.

Case Studies: Sub-selective Quantization

With the theoretical guarantee, we are now ready to apply sub-selective multiplication on existing quantization methods, i.e. PCAQ (Wang, Kumar, and Chang 2012), ITQ (Gong et al. 2013). A common initial step of them is PCA projection (e.g. Alg. 1 and Alg. 2). The time complexity for matrix multiplication $X^T X$ is $O(nd^2))$ when $d < n$. For large n, this step could take up 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 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 $O(md^2)$, where $m \ll n$ in large dataset. For ITQ, the learning process includes dozens of iterations to find rotation matrix R (Alg. 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 sub-selection version of B and V in Alg. 2 respectively. The time complexity of compute R is reduced from $O(nc^2)$ to $O(mc^2)$.

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's omitted because of the page limits. Complexity of original ITQ is $O(nd^2 + (p+1)nc^2)$. In contrast, complexity of ITQ-SS is reduced to $O(md^2 + pmc^2 + nc^2)$. 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 Othorgonal Gaussian Random matrix;
for $k = 1$ to $p$ do
    uniformly randomly generate $\Omega \subset [1:n]$;
    compute $V_\Omega$;
    $B_\Omega = sgn(V_\Omega R)$;
    $[S, \Lambda, \hat{S}] = svd(B_\Omega^T V_\Omega)$;
    $R = \hat{S} S^T$;
end for
7. $B = sgn(VR)$.
```

Evaluations

Experimental Setting

In this section, we evaluate the Sub-selective Quantization approaches on three public datasets: **CIFAR** (Krizhevsky and Hinton 2009) [1], **MNIST**[2] and **Tiny-1M** (Wang, Kumar, and Chang 2012).

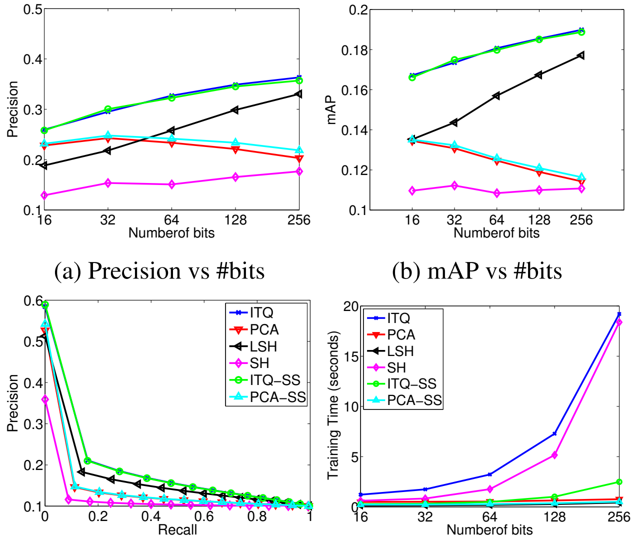
- **CIFAR** consists of 60K 32 × 32 color images that have been manually labelled to ten categories. Each category contains 6K samples. Each image in CIFAR is assigned to one mutually exclusive class label and represented by a 512-dimensional GIST feature vector (Oliva and Torralba 2001).
- **MNIST** consists of 70K samples of 784-dimensional feature vector associated with digits from '0' to '9'. The true neighbours are defined semantic neighbours based on the associated digit labels.
- **Tiny-1M** consists of one million images. Each image is represented by a 384-dimensional GIST vector. Since manually labels are not available on Tiny-1M, Euclidean neighbours are computed and used as ground truth of nearest neighbour search.

We compare proposed methods **PCAQ-SS** and **ITQ-SS** with their corresponding unaccelerated methods **PCAQ** (Wang, Kumar, and Chang 2012) and **ITQ** (Gong et al. 2013). We also compare our methods to two baseline methods that follow similar quantization scheme $B = sgn(X\tilde{W})$: **1) LSH** (Andoni and Indyk 2006), $\tilde{W}$ is a Gaussian random matrix; **2) SH** (Weiss, Torralba, and Fergus 2008), 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.

Two types of evaluation are conducted following (Gong et al. 2013). First, semantic consistency of codes is evaluated for different methods while class labels are used as ground truth. We report 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 use the generated codes for nearest neighbour search, where Euclidean neighbours are used as ground truth. This experiment is conducted on Tiny-1M dataset. We report the three measures: **average precision of top 5% ranked images** for each query and **training time**. For both types of evaluation, the query algorithm and corresponding structure of binary code are the same, so **testing time** are exactly the same for all the methods except SH. Hence, it's omitted from the results. For the limit of page length, only parts of results are presented while the rest are put in the supplementary materials. All our experiments were conducted on a desktop computer with a 3.4GHz Intel Core i7 and 12GB RAM.

Results on CIFAR Dataset

The CIFAR dataset 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 randomly generate our sub-selective matrix Ω with cardinality equals to $1/40$ of number of data points, i.e. $|\Omega| = m = n/40$.



(a) Precision vs #bits

(b) mAP vs #bits

(c) Recall precision @64 bits

(d) Training time vs #bits

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

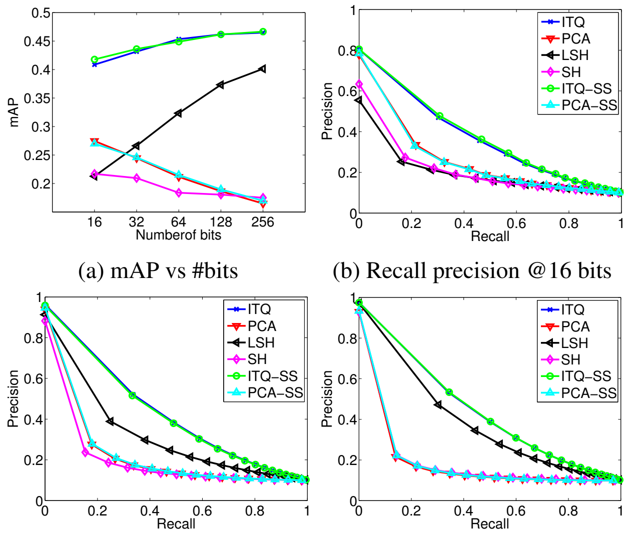
Figure 1(a) and Figure 1(b) show complete precision of top 100 ranked images and mean average precision (mAP) over 1K query images for different number of bits. Figure 1(c) shows recall-precision curse of 64 bits code. For these three metrics, ITQ and ITQ-SS have the best performance. Both sub-selective methods (PCAQ-SS and ITQ-SS) preserve the performance of original methods (i.e. PCAQ and ITQ). Our results indicate that sub-selection preserve semantic consistency of original coding method. Figure 1(d) shows the training time of the two methods. Our method is about **4 to 8 times faster** than ITQ (Gong et al. 2013). 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 reduce the dimension of the problem from a function of n to that of m, where $m \ll n$. These results validate the benefits of sub-selection to preserve the performance of original method with far less training cost.

Results on MNIST Dataset

The MNIST dataset is splited into two subsets: 69K samples as a training set and 1K samples as a query set. While CIFAR dataset evaluates the performance of sub-selective quantization on complex visual features, MNIST evaluates that on raw pixel features. Similar to the previous experiment on CIFAR, we uniformly randomly generate our sub-selective matrix Ω with cardinality equals to $1/40$ of number of datapoints, i.e. $|\Omega| = m = n/40$. Figure 2(b) to Figure 2(d) shows three recall-precision curves of Hamming ranking over 1K images corresponding to 16, 64 and 256 bits code. In all cases, the two curves of ITQ and proposed ITQ-

[1] http://www.cs.toronto.edu/ kriz/cifar.html

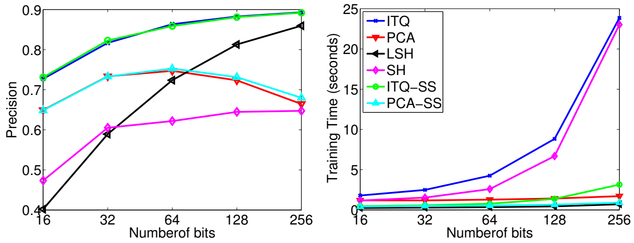
[2] http://yann.lecun.com/exdb/mnist/



(a) mAP vs #bits (b) Recall precision @16 bits

(c) Recall precision @64 bits (d) Recall precision @256 bits

Figure 2: Results on MNIST. All the subfigures share the same set of legends.



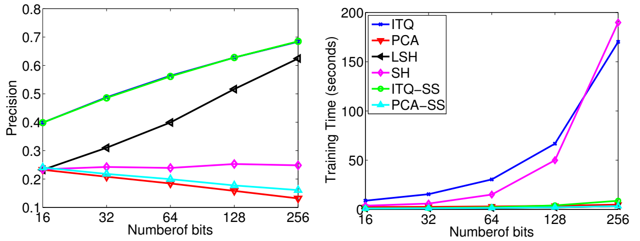
(a) Precision vs #bits (b) Training time vs #bits

Figure 3: The results on MNIST. All the subfigures share the same set of legends.

SS are almost overlapping in all segments. Same trend can be seen for PCAQ and PCAQ-SS. Figure 2(a) and Figure 3(a) show complete precision of top 100 ranked images and mean average precision (mAP) over 1K query images for different number of bits. The difference between ITQ and proposed ITQ-SS are almost negligible. The results confirm the trends seen in Figure 3(a). Figure 3(b) shows the training time of the two methods. Our method is about **3 to 8 times faster** than ITQ. The results of performance and training time are consistent with results on CIFAR. These results again validate the benefits of sub-selection.

Results on Tiny-1M Dataset

For experiment without labelled groundtruth, a separate subset of 2K images of 80 million images are used as the test set while another one million images are used as the training set. We uniformly randomly generate our sub-selective matrix Ω with cardinality equals to $1/1000$ of number of data points, i.e. $|\Omega| = m = n/1000$. Figure 4(a) shows complete precision of top 5% ranked images and mean average precision (mAP) over 1K query images for different number of bits. The difference between sub-selective methods (i.e. PCAQ-SS, ITQ-SS) and their counterparts (i.e. PCAQ, ITQ) are less than 1%. Figure 4(b) shows the training time of the two methods. The ITQ-SS have achieved even bigger speed advantage, which is about **10 to 30 times faster** than ITQ. This is because the larger dataset samples are more redundant, making it possible to use smaller portion of data.



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

Figure 4: The results on Tiny-1M. All the subfigures share the same set of legends.

Discussion and Conclusion

All of the experimental results have verified the benefits of the sub-selective quantization technique whose parameters can be automatically learned from a subset of the original dataset. The proposed PCAQ-SS and ITQ-SS methods have achieved 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 datasets, *e.g.*, 10 to 30 times faster on Tiny-1M. Hence, for larger datasets good quantization quality can be achieved with an even lower sampling ratio.

One may notice that the speed-up ratio is not as 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 dataset. The latter inevitably involves the operations upon the whole dataset, which costs a considerable number of matrix multiplications. In fact, this is one single step requiring matrix multiplications, thus enabling an easy acceleration by using parallel or distributed computing techniques. We will leave this problem to future work.

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 the existing methods. The extensive image retrieval results on large image corpora with size up to one million further empirically verified the speed gain of sub-selective quantization.

References

Andoni, A., and Indyk, P. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. FOCS*.

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9(6):717–772.

Ge, T.; He, K.; Ke, Q.; and Sun, J. 2013. Optimized product quantization for approximate nearest neighbor search. In *Proc. CVPR*.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence* 35(12):2916–2929.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Huang, J.; Zhang, S.; Li, H.; and Metaxas, D. 2011. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding* 115(12):1610–1622.

Huang, J.; Huang, X.; and Metaxas, D. 2009. Learning with dynamic group sparsity. In *Computer Vision, 2009 IEEE 12th International Conference on*, 64–71. IEEE.

Huang, J.; Zhang, S.; and Metaxas, D. 2011. Efficient mr image reconstruction for compressed mr imaging. *Medical Image Analysis* 15(5):670–679.

Jegou, H.; Douze, M.; and Schmid, C. 2011. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Analysis and Machine Intelligence* 33(1):117–128.

Kong, W., and Li, W.-J. 2012. Isotropic hashing. In *NIPS 25*.

Korman, S., and Avidan, S. 2011. Coherency sensitive hashing. In *Proc. ICCV*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

Kulis, B., and Darrell, T. 2009. Learning to hash with binary reconstructive embeddings. In *NIPS 22*.

Kulis, B., and Grauman, K. 2012. Kernelized locality-sensitive hashing. *IEEE Trans. Pattern Analysis and Machine Intelligence* 34(6):1092–1104.

Kulis, B.; Jain, P.; and Grauman, K. 2009. Fast similarity search for learned metrics. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31(12):2143–2157.

Li, Y.; Chen, C.; and Huang, J. 2014. Transformation-invariant collaborative sub-representation. In *22th International Conference on Pattern Recognition*. IEEE.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *Proc. ICML*.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012. Supervised hashing with kernels. In *Proc. CVPR*.

McDiarmid, C. 1989. On the method of bounded differences. *Surveys in combinatorics* 141(1):148–188.

Mu, Y.; Shen, J.; and Yan, S. 2010. Weakly-supervised hashing in kernel space. In *Proc. CVPR*.

Norouzi, M., and Blei, D. M. 2011. Minimal loss hashing for compact binary codes. In *Proc. ICML*.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42(3):145–175.

Raginsky, M., and Lazebnik, S. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS 22*.

Strecha, C.; Bronstein, A. M.; Bronstein, M. M.; and Fua, P. 2012. Ldahash: Improved matching with smaller descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence* 34(1):66–78.

Torralba, A.; Fergus, R.; and Freeman, W. T. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30(11):1958–1970.

Torralba, A.; Fergus, R.; and Weiss, Y. 2008. Small codes and large image databases for recognition. In *Proc. CVPR*.

Wang, J.; Kumar, S.; and Chang, S.-F. 2012. Semi-supervised hashing for large-scale search. *IEEE Trans. Pattern Analysis and Machine Intelligence* 34(12):2393–2406.

Weiss, Y.; Fergus, R.; and Torralba, A. 2012. Multidimensional spectral hashing. In *Proc. ECCV*.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *NIPS 21*.

Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *Proc. ICCV*.