Batch-Orthogonal Locality-Sensitive Hashing for Angular Similarity

Jianqiu Ji, Shuicheng Yan, *Senior Member, IEEE*, Jianmin Li, Guangyu Gao, Qi Tian, *Senior Member, IEEE*, and Bo Zhang

Abstract—Sign-random-projection locality-sensitive hashing (SRP-LSH) is a widely used hashing method, which provides an unbiased estimate of pairwise angular similarity, yet may suffer from its large estimation variance. We propose in this work batch-orthogonal locality-sensitive hashing (BOLSH), as a significant improvement of SRP-LSH. Instead of independent random projections, BOLSH makes use of batch-orthogonalized random projections, i.e., we divide random projection vectors into several batches and orthogonalize the vectors in each batch respectively. These batch-orthogonalized random projections partition the data space into regular regions, and thus provide a more accurate estimator. We prove theoretically that BOLSH still provides an unbiased estimate of pairwise angular similarity, with a smaller variance for any angle in $(0, \pi)$, compared with SRP-LSH. Furthermore, we give a lower bound on the reduction of variance. The extensive experiments on real data well validate that with the same length of binary code, BOLSH may achieve significant mean squared error reduction in estimating pairwise angular similarity. Moreover, BOLSH shows the superiority in extensive approximate nearest neighbor (ANN) retrieval experiments.

Index Terms—Sign-random-projection, locality-sensitive hashing, angular similarity, approximate nearest neighbor search

1 Introduction

Locality-sensitive hashing (LSH) aims to hash similar data samples to the same hash code with high probability [1], [2]. Based on the locality-sensitive property, a fundamental usage of LSH is to generate sketches, or signatures, or fingerprints, for reducing storage space while approximately preserving the pairwise similarity. These sketches or signatures can be used for higher-level applications, e.g., clustering [3], [4], near-duplicate detection [5], [6], [7], [8]. Moreover, LSH can further be used for efficient approximate nearest neighbor (ANN) search [2], which is one of its most important applications. We can index the hash code in an efficient way, i.e., in hash tables, to enable efficient search for similar data samples to a query.

Binary LSH is a special kind of LSH that generates binary codes. It approximates a certain distance or similarity of two data samples by computing the Hamming distance between the corresponding binary codes. The advantages of binary LSH are two-fold: on the one hand, computing Hamming distance involves mainly bitwise operations, and it is much faster than directly computing other distances, e.g., euclidean and cosine distances, which require heavy arithmetic operations; on the other hand, the storage is substantially reduced due to the use of compact binary codes. In large-scale applications [3], [7], [9], [11], e.g., near-duplicate image or document detection, object and scene recognition, large-scale clustering, etc., we are often confronted with intensive computing of pairwise distances or similarities, then binary LSH may act as a scalable solution.

Sign-random-projection locality-sensitive hashing (SRP-LSH) [12] is an important binary LSH method, which is widely used and extensively studied. The Hamming distance between two codes of SRP-LSH provides an unbiased estimate of the pairwise angular similarity. For many kinds of data represented by vectors, the natural pairwise similarity is only related to the angle between the data, e.g., the normalized bag-of-words representation for documents, images, and videos, and the normalized histogram-based local features like SIFT [13]. Specifically, SIFT descriptors are usually normalized to unit length, to gain further robustness against various lighting conditions. Thus, the resulting normalized SIFT descriptors are points lying on the unit sphere in $\mathbb{R}^{128}$. In these cases, angular similarity can serve as a similarity measurement, and SRP-LSH can be used as a hashing method. For example, SRP-LSH is used for high speed noun clustering [3], near-duplicate web document detection [7], similarity search in large-scale database [12], and it is the basic building block of many other binary embedding or hashing algorithms [14], [15], [16].

Although SRP-LSH is widely used, it may suffer from the large variance of its estimation. In our previous work [17], of which this work is a journal extension, we proposed batch-orthogonal Locality-Sensitive Hashing (BOLSH), as an improvement over SRP-LSH. Instead of independent

- *J. Ji, J. Li, and B. Zhang are with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P.R. China. E-mail: jijq10@mails.tsinghua.edu.cn, {lijianmin, dcszb}@mail.tsinghua.edu.cn.*
- *G. Gao is with the School of Software, Beijing Institute of Technology, Beijing 100081, P.R. China. E-mail: guangyugao@bit.edu.cn.*
- *S. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576. E-mail: eleyans@nus.edu.sg.*
- *Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, One UTSA Circle, University of Texas at San Antonio, San Antonio, TX 78249-1604. E-mail: qi.tian@utsa.edu.*

Manuscript received 1 Mar. 2013; revised 14 Feb. 2014; accepted 16 Mar. 2014. Date of publication 3 Apr. 2014; date of current version 10 Sept. 2014. Recommended for acceptance by S. Avidan.
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.2014.2315806

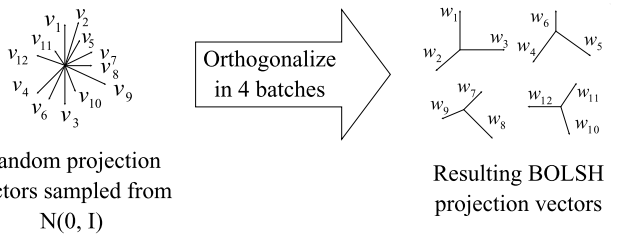


Fig. 1. An illustration of 12 BOLSH projection vectors $\{w_i\}$ generated by orthogonalizing independent random projection vectors $\{v_i\}$ in four batches.

random projections, BOLSH makes use of batch-orthogonalized random projection vectors, as illustrated in Fig. 1. It is proven in [17] that BOLSH also provides an unbiased estimate of pairwise angular similarity, and has a smaller variance than SRP-LSH when the angle to estimate is in $(0, \pi/2]$.

As the journal extension version of our previous work [17], this work gives much stronger theoretical results. We further provide theoretical justifications that the variance of BOLSH is strictly smaller than that of SRP-LSH, for any angle in $(0, \pi)$. Moreover, we prove a lower bound on the reduction of variance.

The proposed BOLSH method is closely related to many recently proposed Principal Component Analysis (PCA)-style learning-based hashing methods, which learn orthogonal projections. Although BOLSH is purely probabilistic and data-independent, the model of orthogonal random projection together with its theoretical justifications can help gain more insights and a better understanding of these learning-based hashing methods. Furthermore, since theoretical analysis and experiments both show that BOLSH approximates the angle between two vectors more accurately, BOLSH, in replace of SRP-LSH, can be used in various applications requiring massive angle-related computations, e.g., dot product [18], angular similarity, cosine similarity, euclidean distance. Its potential applications include approximate nearest neighbor search, near-duplicate detection, clustering and so on.

In Section 2, we introduce the related work. In Section 3, we first review SRP-LSH, then describe the proposed BOLSH. In Section 4, we give theoretical justifications to show that BOLSH has a smaller variance than SRP-LSH, and we also provide a lower bound for the variance reduction. In Section 5, we conduct experiments to demonstrate the effectiveness of the proposed BOLSH, and verify the theoretical analysis.

2 Related Work

Apart from SRP-LSH, there are various LSH methods for different similarities, e.g., p-stable-distribution LSH [19] for ℓ_p-distance when $p \in (0, 2]$. Bit-sampling LSH methods [1], [2] for Hamming distance and ℓ_1-distance, min-wise hash [4], [6] for Jaccard similarity.

All these LSH methods are probabilistic and data-independent. Due to their simplicity, they are easy to be integrated as a module in more complicated algorithms involving pairwise distance or similarity computation, or similarity search. Besides approximate nearest neighbor search, LSH methods are widely used in near-duplicate detection [5], [6], [7], [8], clustering [3], [4], document fingerprinting [20], image retrieval [21], [22], object indexing [23], and so on. Particularly, Georgescu et al. [24] made use of LSH to reduce the computation complexity of adaptive mean-shift, for clustering in high dimensions. Dean et al. [18] used LSH to accelerate the dot product computation, enabling fast and accurate detection of 100,000 object classes on a single machine.

New probabilistic data-independent methods for improving these original LSH methods have been proposed recently. Andoni and Indyk [25] proposed a near-optimal LSH method for euclidean distance. B-bit minwise hash [26] improves the original min-hash in terms of compactness. Li et al. [11] showed that b-bit minwise hash can be integrated in linear learning algorithms for large-scale learning tasks. Shift-invariant kernel hashing [27] is a probabilistic data-independent method for shift-invariant kernels. Li et al. [28] proposed very sparse random projections for accelerating random projections and SRP.

Data-dependent hashing methods have also been extensively studied. While LSH is used in various applications, data-dependent hashing is specifically designed for approximate nearest neighbor search. The advantage of data-dependent methods is that they learn the proper hash functions such that these hash functions partition the data space in a proper way according to the data distribution. Since they can fit the data, they usually perform better in approximate nearest neighbor search compared with the purely probabilistic data-independent LSH methods. However, the disadvantage is that there is no theoretical guarantee for these methods on how well the hash functions would fit the data. The main reason is that the objective functions of these methods are non-convex, or even not continuous, and thus there is no global guarantee for these methods. Furthermore, when the data is of high dimension, and the distribution is complicated, learning the hash functions may not be scalable. Besides, as mentioned above, the application of data-dependent hashing is limited to approximate nearest neighbor search.

Data-dependent hashing methods can be categorized into unsupervised, semi-supervised and supervised methods. Spectral hashing [29], anchor graph hashing [30], iterative quantization [31], anti-sparse coding [32], K-means hashing [33] are data-dependent unsupervised methods. There are also a bunch of works devoted to supervised and semi-supervised hashing methods [14], [34], [35], [36], [37], [38], [39], [40], [41], which try to capture not only the geometry of the original data, but also the semantic relations.

There are many extensions based on the hashing methods mentioned above, towards scalable similarity search. For example, Lv et al. [42] proposed a multi-probe scheme, to improve the search quality of p-stable-distribution LSH while reducing the number of hash tables. Satuluri et al. [43] proposed Bayesian LSH for fast pruning away a substantial amount of false positive candidates during post-processing. Norouzi et al. [44] made use of multi-index structure to enable fast exact K-nearest neighbor search in Hamming space.

Before this work, the technique of using row-orthogonal or column-orthogonal random projection matrix has been

proposed in several papers. Jégou et al. [45] proposed Hamming embedding method, which uses row-orthogonal random projection matrix to produce binary signatures for refining the matching based on visual words. The function for producing binary signatures is different from the hash function of SRP-LSH and the proposed BOLSH in this paper. Furthermore, since the technique of batch-orthogonalization proposed in this work provides a general framework, BOLSH allows both small code length and long code length which exceeds the data dimension, while Hamming embedding only produces small codes.

The use of column-orthogonal random projection matrix appeared in [32], and the method is called LSH+frame. The experiment shows that orthogonalizing the columns of the random projection matrix would achieve better result in approximate nearest neighbor search. However, compared with BOLSH, this method orthogonalizes the random matrix in a perpendicular way, which can only generate binary codes with long code length that is not smaller than the data dimension, and thus it is completely different from the proposed BOLSH in this work.

Though these prior works [32], [45] already show the interest of orthogonalizing the random projection matrix in some ways, they are all based on empirical observations. This work formally justifies the superiority of using batch-orthogonalized random projections in constructing hash functions. In particular, we provide theoretical justification to guarantee that the hash functions of BOLSH, which are constructed by batch-orthogonalized random projections, satisfy locality-sensitive property. Furthermore, we show that the variance of BOLSH is smaller than that of SRP-LSH.

Several data-dependent hashing methods [29], [31], [38] proposed recently learn orthogonal projection vectors. Based on the formulation in [29], [38], Gong and Lazebnik [31] proposed iterative quantization method, which first conducts dimension reduction via PCA, then iteratively optimizes and searches for a rotation matrix to minimize the quantization loss.

Although the proposed BOLSH is a probabilistic data-independent method, the orthogonal random projection model and the theoretical analysis for the use of orthogonalization can help gain a better understanding of these learning-based hashing methods.

3 Batch-Orthogonal Locality-Sensitive Hashing for Angular Similarity

3.1 Sign-Random-Projection Locality-Sensitive Hashing: A Review

Sign-random-projection locality-sensitive hashing [12] is a widely used locality-sensitive hashing method for angular similarity, which embeds real vectors into Hamming space. The resulting pairwise Hamming distance provides an unbiased estimate of the pairwise angular similarity between the original data pair [12], [46]. Angular similarity is defined as follows:

Definition 1. $sim(a, b) = 1 - \theta_{a,b}/\pi$.

Here $\theta_{a,b} = \arccos(\frac{\langle a,b\rangle}{\|a\|\|b\|}) \in [0, \pi]$ is the angle between a and b, where $\langle a, b\rangle$ denotes the inner product of a and b, and $\|\cdot\|$ denotes the ℓ_2-norm of a vector.

Formally, in a d-dimensional data space, let v denote a random vector sampled from the normal distribution $\mathcal{N}(0, I_d)$, and x denote a data sample, then an SRP-LSH function is defined as

Definition 2. $h_v(x) = sgn(v^T x)$.

Here the sign function $sgn(\cdot)$ is defined as

$$sgn(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0. \end{cases}$$

Given two data samples a and b, it is proven [46] that

$$Pr(h_v(a) \neq h_v(b)) = \frac{\theta_{a,b}}{\pi}. \tag{1}$$

We call this the *locality-sensitive* property.

By independently sampling K d-dimensional vectors $v_1, \ldots, v_K$ from the normal distribution $\mathcal{N}(0, I_d)$, we may define a binary-vector-valued function $h(x) = (h_{v_1}(x), h_{v_2}(x), \ldots, h_{v_K}(x))$, which concatenates K SRP-LSH functions and thus produces K-bit codes. Then by the locality-sensitive property (1), it is easy to prove that

$$\mathbb{E}[d_{Hamming}(h(a), h(b))] = \frac{K\theta_{a,b}}{\pi} = C\theta_{a,b}. \tag{2}$$

where $C = K/\pi$.

This reveals the relation between Hamming distance and angular similarity. That is, the expectation of the Hamming distance between the binary hash codes of two given data samples a and b is an unbiased estimate of their angle $\theta_{a,b}$, up to a constant scale factor $C = K/\pi$. Thus SRP-LSH provides an unbiased estimate of angular similarity.

Since $\{h_{v_1}(x), h_{v_2}(x), \ldots, h_{v_K}(x)\}$ are independent SRP functions, $d_{Hamming}(h(a), h(b))$ follows a binomial distribution, i.e., $d_{Hamming}(h(a), h(b)) \sim \mathcal{B}(K, \frac{\theta_{a,b}}{\pi})$, and thus its variance is

$$Var[d_{Hamming}(h(a), h(b))] = \frac{K\theta_{a,b}}{\pi}\left(1 - \frac{\theta_{a,b}}{\pi}\right). \tag{3}$$

This implies that the variance of the normalized Hamming distance $d_{Hamming}(h(a), h(b))/K$, i.e., $Var[d_{Hamming}(h(a), h(b))/K]$,satisfies

$$Var[d_{Hamming}(h(a), h(b))/K] = \frac{\theta_{a,b}}{K\pi}\left(1 - \frac{\theta_{a,b}}{\pi}\right). \tag{4}$$

Though being widely used, SRP-LSH may suffer from the large variance of its estimation, leading to large estimation error. Generally we need a substantially long code length to accurately approximate the angular similarity [35], [36], [47].

Since independent random vectors do not partition the input space in a regular manner, the resulting binary code may be less informative as it seems, and may even contain many redundant bits. To tackle this problem, an intuitive idea would be to orthogonalize the random vectors. However, once being orthogonalized, the projection vectors are

no longer independently sampled. Moreover, it remains unclear whether the resulting Hamming distance still provides an unbiased estimate of the angle $\theta_{a,b}$, and what its variance will be. In Section 4 we will give answers with theoretical justifications to these two questions.

In the next section, based on the above intuitive idea, we propose batch-orthogonal locality-sensitive hashing method. We provide theoretical guarantees that after orthogonalizing the random projection vectors in batches, we still get an unbiased estimate of angular similarity, yet with a smaller variance, for any angle $\theta_{a,b} \in (0, \pi)$. Thus the resulting binary code is more informative. Experiments on real data show the effectiveness of BOLSH, which with the same length of binary code may achieve as much as 30 percent mean squared error (MSE) reduction compared with SRP-LSH in estimating angular similarity on real data. Moreover, BOLSH shows its effectiveness in approximate nearest neighbor retrieval experiments.

3.2 Batch-Orthogonal Locality-Sensitive Hashing

The proposed BOLSH is based on SRP-LSH. When the code length K satisfies $1 < K \leq d$, where d is the dimension of data space, we can orthogonalize N $(1 \leq N \leq \min(K, d) = K)$ of the random vectors sampled from the normal distribution $\mathcal{N}(0, I_d)$. The orthogonalization procedure is the QR decomposition process.

After orthogonalization, the resulting N vectors are no longer independently sampled, thus we group their corresponding bits together as an *N-batch*. We call N the *batch size*.

However, when the code length $K > d$, it is impossible to orthogonalize all K vectors. Without loss of generality, assume that $K = N \times L$, and $1 \leq N \leq d$, then we can perform the QR decomposition process L times to orthogonalize them in L batches. Formally, K random vectors $\{v_1, v_2 \ldots, v_K\}$ are independently sampled from the normal distribution $\mathcal{N}(0, I_d)$, and then are divided into L batches with N vectors each. Perform the QR decomposition process to these L batches of N vectors respectively, and we get $K = N \times L$ projection vectors $\{w_1, w_2 \ldots, w_K\}$. This results in K BOLSH functions $(h_{w_1}, h_{w_2} \ldots, h_{w_K})$, where h_{w_i} is defined as

Definition 3. $h_{w_i}(x) = sgn(w_i^T x)$.

These K functions produce L N-batches and altogether produce K-bit binary codes. Fig. 1 shows an example of generating 12 BOLSH projection vectors. Algorithm 1 describes the procedure for generating BOLSH projection vectors.

When the batch size $N = 1$, BOLSH degenerates to SRP-LSH. In other words, SRP-LSH is a special case of BOLSH.

Note that the definition of the BOLSH function (Definition 3) shares the same form as that of the SRP-LSH function (Definition 2). The key difference is the choice of the projection vectors. Instead of independent random projections, BOLSH uses random projections that are orthogonalized in batches.

Also note that the QR decomposition process is not the only way of producing a set of orthogonal vectors uniformly at random. Performing singular value decomposition to a random matrix with i.i.d. standard normal elements also achieves this goal. Although the proofs of some of the theoretical results in this paper depend on QR decomposition, we expect that the same theoretical results can be achieved by using the singular value decomposition method.

There are several ways of computing the QR decomposition, such as the Gram-Schmidt process, Householder transformations and Givens rotations.

The algorithm can be easily extended to the case where the code length K is not a multiple of the batch size N. In fact one can even use variable batch size N_i, as long as $1 \leq N_i \leq d$.

With the same code length, BOLSH has the same running time $O(Kd)$ as SRP-LSH in on-line processing, i.e., generating binary codes when applying to data.

Algorithm 1 Generating Batch-Orthogonal Locality-Sensitive Hashing Projection Vectors

Input: Data space dimension d, batch size $1 \leq N \leq d$, number of batches $L \geq 1$, resulting code length $K = N \times L$.

Generate a random matrix H with each element being sampled independently from the normal distribution $\mathcal{N}(0, 1)$. Denote $H = [v_1, v_2, ..., v_K]$.

for $i = 0$ **to** $L - 1$ **do**
  Denote $H_i = [v_{iN+1}, v_{iN+2}, ..., v_{(i+1)N}]$.
  Compute the QR decomposition of H_i, such that $H_i = Q_i R_i$.
  Take the first N columns of Q_i and denote them by $[w_{iN+1}, w_{iN+2}, ..., w_{(i+1)N}]$.
end for

Output: $\tilde{H} = [w_1, w_2, ..., w_K]$.

4 Theoretical Analysis

In this section we provide theoretical analysis of BOLSH. We show that the Hamming distance between the binary codes produced by BOLSH provides an unbiased estimate of the angle between the corresponding two vectors, with a smaller variance. Moreover, we prove a lower bound on the reduction of variance.

4.1 Unbiased Estimate

In this section we show that BOLSH provides an unbiased estimate of the angle $\theta_{a,b}$ of $a, b \in \mathbb{R}^d$.

The intuition is that, after orthogonalization, each projection vector still follows an isotropic distribution, i.e., it points to any direction in $\mathbb{R}^d$ with equal probability density, and thus the probability that its corresponding hyperplane partitions two vectors is proportional to the angle (that equals $\theta_{a,b}/\pi$). Therefore with the linearity of expectation, the Hamming distance provides an unbiased estimate of the angle.

First we list the important lemmas needed for the proof. Lemmas 3 and 4 state that after orthogonalization, each random vector still follows an isotropic distribution. Lemmas 1 and 2 show that a random vector that follows an isotropic

distribution satisfies the locality-sensitive property. Then Theorem 1 and Corollary 1 prove that the Hamming distance between BOLSH binary codes provides an unbiased estimate of the angle between the corresponding two given vectors.

Lemma 1. *([46], Lemma 3.2) Let $\mathcal{S}^{d-1}$ denote the unit sphere in $\mathbb{R}^d$. Given a random vector v uniformly sampled from $\mathcal{S}^{d-1}$, we have $Pr[h_v(a) \neq h_v(b)] = \theta_{a,b}/\pi$.*

Lemma 2. *If $v \in \mathbb{R}^d$ follows an isotropic distribution, then $\overline{v} = v/\|v\|$ is uniformly distributed on $\mathcal{S}^{d-1}$.*

This lemma can be proven by the definition of isotropic distribution, and we omit the details here.

Lemma 3. *Given k vectors $v_1, \ldots, v_k \in \mathbb{R}^d$, which are sampled i.i.d. from the normal distribution $\mathcal{N}(0, I_d)$, and span a subspace S_k, let P_{S_k} denote the orthogonal projection onto S_k, then P_{S_k} is a random matrix uniformly distributed on the Grassmann manifold $G_{k,d-k}$.*

This lemma can be proven by applying the Theorems 2.2.1 (iii) and 2.2.2 (iii) in [48].

Lemma 4. *If P is a random matrix uniformly distributed on the Grassmann manifold $G_{k,d-k}$, $1 \leq k \leq d$, and $v \sim \mathcal{N}(0, I_d)$ is independent of P, then the random vector $\tilde{v} = Pv$ follows an isotropic distribution.*

From the uniformity of P on the Grassmann manifold and the property of the normal distribution $\mathcal{N}(0, I_d)$, we can get this result directly. We give a sketch of proof below.

Proof. We can write $P = UU^T$, where the columns of $U = [u_1, u_2, \ldots, u_k]$ constitute the orthonormal basis of a random k-dimensional subspace. Since the standard normal distribution is two-stable [19], $\hat{v} = U^T v = [\hat{v_1}, \hat{v_2}, \ldots, \hat{v_k}]^T$ is a $\mathcal{N}(0, I_k)$-distributed vector, where each $\hat{v_i} \sim \mathcal{N}(0, 1)$, and it is easy to verify that $\hat{v}$ is independent of U. Therefore $\tilde{v} = Pv = U\hat{v} = \Sigma_{i=1}^k \hat{v_i} u_i$. Since $u_i, \ldots, u_k$ can be the orthonormal basis of any k-dimensional subspace with equal probability density, and $\{\hat{v_1}, \hat{v_2}, \ldots, \hat{v_k}\}$ are i.i.d. $\mathcal{N}(0, 1)$ random variables, $\tilde{v}$ follows an isotropic distribution. □

Theorem 1. *Given N i.i.d. random vectors $v_1, v_2, \ldots, v_N \in \mathbb{R}^d$ sampled from the normal distribution $\mathcal{N}(0, I_d)$, where $1 \leq N \leq d$, perform the QR decomposition process to them and produce N orthogonalized vectors $w_1, w_2, \ldots, w_N$, then for any two data vectors $a, b \in \mathbb{R}^d$, by defining N indicator random variables $X_1^{a,b}, X_2^{a,b}, \ldots, X_N^{a,b}$ as*

$$X_i^{a,b} = \begin{cases} 1, & h_{w_i}(a) \neq h_{w_i}(b) \\ 0, & h_{w_i}(a) = h_{w_i}(b) \end{cases}$$

we have $\mathbb{E}[X_i^{a,b}] = \theta_{a,b}/\pi$, for any $1 \leq i \leq N$. Note that for simplicity, from now on we omit the explicit dependency on the data pair a and b of the symbols $X_1^{a,b}, X_2^{a,b}, \ldots, X_N^{a,b}$, and just denote them by $X_1, X_2, \ldots, X_N$.

Proof. Denote S_{i-1} the subspace spanned by $\{w_1, \ldots, w_{i-1}\}$, and the orthogonal projection onto its orthogonal complement as $P_{S_{i-1}}^{\perp}$. Then $w_i = P_{S_{i-1}}^{\perp} v_i$. Denote $\overline{w} = w_i/\|w_i\|$.

For any $1 \leq i \leq N$, $\mathbb{E}[X_i] = Pr[X_i = 1] = Pr[h_{w_i}(a) \neq h_{w_i}(b)] = Pr[h_{\overline{w}}(a) \neq h_{\overline{w}}(b)]$. For $i = 1$, by Lemmas 1 and 2, we have $Pr[X_1 = 1] = \theta_{a,b}/\pi$.

For any $1 < i \leq N$, consider the distribution of w_i. By Lemma 3, $P_{S_{i-1}}$ is a random matrix uniformly distributed on the Grassmann manifold $G_{i-1,d-i+1}$, thus $P_{S_{i-1}}^{\perp} = I - P_{S_{i-1}}$ is uniformly distributed on $G_{d-i+1,i-1}$. Since $v_i \sim \mathcal{N}(0, I_d)$ is independent of $v_1, v_2, \ldots, v_{i-1}$, v_i is independent of $P_{S_{i-1}}^{\perp}$. By Lemma 4, we have that $w_i = P_{S_{i-1}}^{\perp} v_i$ follows an isotropic distribution. By Lemma 2, $\overline{w} = w_i/\|w_i\|$ is uniformly distributed on the unit sphere in $\mathbb{R}^d$. By Lemma 1, $Pr[h_{\overline{w}}(a) \neq h_{\overline{w}}(b)] = \theta_{a,b}/\pi$. □

Corollary 1. *For any batch size N, $1 \leq N \leq d$, assuming that the code length $K = N \times L$, the Hamming distance $d_{Hamming}(h(a), h(b))$ is an **unbiased estimate** of $\theta_{a,b}$, for any two data vectors a and $b \in \mathbb{R}^d$, up to a constant scale factor $C = K/\pi$.*

Proof. Apply Theorem 1 and we have that

$$\mathbb{E}[d_{Hamming}(h(a), h(b))] = L \times \mathbb{E}[\Sigma_{i=1}^N X_i] = L \times \Sigma_{i=1}^N \mathbb{E}[X_i]$$
$$= L \times \Sigma_{i=1}^N \theta_{a,b}/\pi = \frac{K\theta_{a,b}}{\pi} = C\theta_{a,b}.$$ □

4.2 Smaller Variance

In this section we show that the variance of BOLSH is strictly smaller than that of SRP-LSH, for any angle $\theta_{a,b} \in (0, \pi)$.

The intuition is that independent random hyperplanes do not divide the input space in a regular manner. However, orthogonal projection vectors divide the input space into regular regions, thus it is expected that BOLSH has a smaller variance in estimating the angle.

First we need to express the variance of the Hamming distance between BOLSH binary codes. Lemmas 5 and 6 unify all the cross-product terms $E[X_i X_j] = Pr[X_i = 1 \mid X_j = 1]\, Pr[X_j = 1]$ in the variance to $Pr[X_2 = 1 \mid X_1 = 1] Pr[X_1 = 1] = Pr[X_2 = 1 \mid X_1 = 1] \frac{\theta_{a,b}}{\pi} = p_{2,1} \frac{\theta_{a,b}}{\pi}$, where $p_{2,1}$ is defined as follows:

Definition 4. $p_{2,1} = Pr[X_2 = 1 \mid X_1 = 1]$.

Then the variance of BOLSH can be expressed in terms of $\theta_{a,b}$, $p_{2,1}$, K and N, as shown in Theorem 2. The smaller is $p_{2,1}$, the smaller is the variance of BOLSH. Lemma 6 proves that $p_{2,1} < \theta_{a,b}/\pi$ for $\theta_{a,b} \in (0, \pi/2]$, leading to Corollary 2 and Corollary 3, that BOLSH has a smaller variance than SRP-LSH, for $\theta_{a,b} \in (0, \pi/2]$. In Section 4.2.2, Theorem 4 shows that the variance of BOLSH is symmetric about $\theta_{a,b} = \pi/2$, so is the reduction of variance. Thus this completes the proof that BOLSH has a smaller variance, for any angle in $(0, \pi)$. To further study how much the variance reduction is, Theorem 3 in Section 4.2.1 gives an upper bound of $p_{2,1}$, leading to a lower bound on the variance reduction. The formal proofs are as follows.

Lemma 5. *For the random variables $\{X_i\}$ defined in Theorem 1, we have the following equality $Pr[X_i = 1 \mid X_j = 1] = Pr[X_i = 1 \mid X_1 = 1]$, $1 \leq j < i \leq N \leq d$.*

Proof. Without loss of generality, we assume that $\|w_k\| = 1$, for $1 \leq k \leq N$. $Pr[X_i = 1 \mid X_j = 1] = Pr[h_{w_i}(a) \neq h_{w_i}(b) \mid X_j = 1] = Pr[h_{v_i - \Sigma_{k=1}^{i-1} w_k w_k^T v_i}(a) \neq h_{v_i - \Sigma_{k=1}^{i-1} w_k w_k^T v_i}(b) | h_{w_j}(a) \neq h_{w_j}(b)]$. Since $\{w_1, \ldots w_{i-1}\}$ is a uniformly random orthonormal basis of a random subspace uniformly distributed

on Grassmann manifold, by exchanging the index j and 1, we have $Pr[X_i = 1 \mid X_j = 1] = Pr[h_{v_i - \Sigma_{k=1}^{i-1} w_k w_k^T v_i}(a) \neq h_{v_i - \Sigma_{k=1}^{i-1} w_k w_k^T v_i}(b) | h_{w_1}(a) \neq h_{w_1}(b)] = Pr[X_i = 1 \mid X_1 = 1]$. □

Lemma 6. *For $\{X_i\}$ defined in Theorem 1, we have $Pr[X_i = 1 \mid X_j = 1] = Pr[X_2 = 1 \mid X_1 = 1]$, $1 \leq j < i \leq N \leq d$. Given $\theta_{a,b} \in (0, \frac{\pi}{2}]$, we have $Pr[X_2 = 1 \mid X_1 = 1] < \frac{\theta_{a,b}}{\pi}$.*

Please refer to Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2315806, for the proof.

With Lemmas 5 and 6, we can express the variance of BOLSH in terms of K, N, $\theta_{a,b}$, and $p_{2,1}$, as in the following Theorem 2:

Theorem 2. *Given two vectors $a, b \in \mathbb{R}^d$ and random variables $\{X_i\}$ defined as in Theorem 1, denote $p_{2,1} = Pr[X_2 = 1 \mid X_1 = 1]$ as in Definition 4, and $S_X = \Sigma_{i=1}^N X_i$ which is the Hamming distance between the N-batches of a and b, for $1 < N \leq d$. Denote $Var[BOLSH_{\theta_{a,b},N,K}]$ as the variance of the Hamming distance produced by BOLSH, where $K = N \times L$ is the code length. Then $Var[BOLSH_{\theta_{a,b},N,N}] = Var[S_X] = \frac{N\theta_{a,b}}{\pi} + N(N-1)\frac{p_{2,1}\theta_{a,b}}{\pi} - (\frac{N\theta_{a,b}}{\pi})^2$, and $Var[BOLSH_{\theta_{a,b},N,K}] = L \times Var[BOLSH_{\theta_{a,b},N,N}] = \frac{K\theta_{a,b}}{\pi} + K(N-1)\frac{p_{2,1}\theta_{a,b}}{\pi} - KN(\frac{\theta_{a,b}}{\pi})^2$.*

This theorem is a combination of the original Theorem 2 and the first part of the Corollary 2 in [17]. We rewrite the theorem for clarity.

Proof. By Lemma 6, $Pr[X_i = 1 \mid X_j = 1] = Pr[X_2 = 1 \mid X_1 = 1] = p_{2,1}$ when $1 \leq j < i \leq N$. Therefore $Pr[X_i = 1, X_j = 1] = Pr[X_i = 1 \mid X_j = 1]\, Pr[X_j = 1] = \frac{p_{2,1}\theta_{a,b}}{\pi}$, for any $1 \leq j < i \leq N$. Therefore $Var[S_X] = \mathbb{E}[S_X^2] - \mathbb{E}[S_X]^2 = \Sigma_{i=1}^N \mathbb{E}[X_i^2] + 2\Sigma_{j<i}\mathbb{E}[X_i X_j] - N^2\mathbb{E}[X_1]^2 = \frac{N\theta_{a,b}}{\pi} + 2\Sigma_{j<i} Pr[X_i = 1, X_j = 1] - (\frac{N\theta_{a,b}}{\pi})^2 = \frac{N\theta_{a,b}}{\pi} + N(N-1)\frac{p_{2,1}\theta_{a,b}}{\pi} - (\frac{N\theta_{a,b}}{\pi})^2$.

Since $v_1, v_2, \ldots, v_K$ are independently sampled, and $w_1, w_2, \ldots, w_K$ are produced by orthogonalizing every N vectors, the Hamming distances produced by different N-batches are independent, thus $Var[BOLSH_{\theta_{a,b},N,K}] = L \times Var[BOLSH_{\theta_{a,b},N,N}] = \frac{K\theta_{a,b}}{\pi} + K(N-1)\frac{p_{2,1}\theta_{a,b}}{\pi} - KN(\frac{\theta_{a,b}}{\pi})^2$. □

Corollary 2. *If $K = N_1 \times L_1 = N_2 \times L_2$ and $1 \leq N_2 < N_1 \leq d$, then $Var[BOLSH_{\theta_{a,b},N_1,K}] - Var[BOLSH_{\theta_{a,b},N_2,K}] = \frac{K\theta_{a,b}}{\pi}(N_1 - N_2)(p_{2,1} - \frac{\theta_{a,b}}{\pi}) < 0$, for any $\theta_{a,b} \in (0, \pi/2]$.*

Proof. By Theorem 2, $Var[BOLSH_{\theta_{ab},N_1,K}] = \frac{K\theta_{ab}}{\pi} + K(N_1 - 1)\frac{p_{2,1}\theta_{ab}}{\pi} - KN_1(\frac{\theta_{ab}}{\pi})^2$. By Lemma 6, when $\theta_{ab} \in (0, \pi/2]$, for $N_1 > N_2 > 1$, $0 \leq p_{2,1} < \frac{\theta_{ab}}{\pi}$. Therefore $Var[\, BOLSH_{\theta_{ab},N_1,K}\,] - Var[\, BOLSH_{\theta_{ab},N_2,K}\,] = \frac{K\theta_{ab}}{\pi}(N_1 - N_2)(p_{2,1} - \frac{\theta_{ab}}{\pi}) < 0$. For $N_1 > N_2 = 1$, $Var[BOLSH_{\theta_{ab},N_1,K}] - Var[BOLSH_{\theta_{ab},N_2,K}] = \frac{K\theta_{ab}}{\pi}(N_1 - 1)(p_{2,1} - \frac{\theta_{ab}}{\pi}) < 0$ □

Corollary 2 shows that when fixing the code length K, as the batch size N goes up, the variance of the Hamming distance produced by BOLSH goes down. Thus by Corollary 2, and the fact that SRP-LSH is a special case of BOLSH when the batch size $N = 1$, we can easily show that $Var[SRPLSH_{\theta_{a,b},K}] = Var[BOLSH_{\theta_{a,b},1,K}] > Var[BOLSH_{\theta_{a,b},N,K}]$. This yields the following Corollary 3:

Corollary 3. *Denote $Var[SRPLSH_{\theta_{a,b},K}]$ as the variance of the Hamming distance produced by SRP-LSH, where $K = N \times L$ is the code length and L is a positive integer, $1 < N \leq d$. Then $Var[SRPLSH_{\theta_{a,b},K}] > Var[BOLSH_{\theta_{a,b},N,K}]$, for any $\theta_{a,b} \in (0, \pi/2]$.*

4.2.1 Lower Bound for Variance Reduction

In this section we give a lower bound on the variance reduction. First, the following theorem gives an upper bound of $p_{2,1}$.

Theorem 3. *For any $\theta_{a,b} \in (0, \pi/2)$,*

$$p_{2,1} < \frac{1}{\pi}\int \arccos\frac{\cos\theta_{a,b}}{\sqrt{1 - Z\sin^2\theta_{a,b}}} p(Z)\, dZ < \frac{1}{\pi}\arccos\frac{\cos\theta_{a,b}}{\sqrt{1 - \frac{\sin^2\theta_{a,b}}{d-1}}} < \frac{\theta_{a,b}}{\pi},$$

where $Z \sim Beta(\frac{1}{2}, \frac{d-2}{2})$.

Please refer to Appendix A, available in the online supplemental material, for the proof.

With the upper bound given by Theorem 3, we directly get a lower bound on the reduction of variance, as shown in the following Corollary 4.

Corollary 4. *$Var[SRPLSH_{\theta_{a,b},K}] - Var[BOLSH_{\theta_{a,b},N,K}] = \frac{K\theta_{a,b}}{\pi}(N-1)(\frac{\theta_{a,b}}{\pi} - p_{2,1}) > \frac{K\theta_{a,b}}{\pi^2}(N-1)(\theta_{a,b} - \arccos\frac{\cos\theta_{a,b}}{\sqrt{1 - \frac{\sin^2\theta_{a,b}}{d-1}}}) > 0$, for any $\theta_{a,b} \in (0, \pi/2)$.*

4.2.2 The Variance in $(\pi/2, \pi)$

In this section we show that the behavior of the variance in $(\pi/2, \pi)$ mirrors that in $(0, \pi/2)$, i.e., the variance is symmetric about $\theta_{a,b} = \pi/2$.

Theorem 4. *$Var[BOLSH_{\theta_{a,b},N,K}] = Var[BOLSH_{\pi - \theta_{a,b},N,K}]$, and*

$$Var[SRPLSH_{\theta_{a,b},K}] - Var[BOLSH_{\theta_{a,b},N,K}] = Var[SRPLSH_{\pi-\theta_{a,b},K}] - Var[BOLSH_{\pi-\theta_{a,b},N,K}].$$

Proof. $Var[BOLSH_{\theta_{a,b},N,K}] = Var[d_{Hamming}(h(a), h(b))] = Var[K - d_{Hamming}(h(-a), h(b))] = Var[d_{Hamming}(h(-a), h(b))] = Var[BOLSH_{\pi-\theta_{a,b},N,K}]$. And since $Var[SRPLSH_{\theta_{a,b},K}] = Var[SRPLSH_{\pi-\theta_{a,b},K}]$, the second statement is proven. □

4.2.3 Discussions

Theorem 2 provides a general expression of the variance of BOLSH in terms of $\theta_{a,b}$, $p_{2,1}$, K and N. Thus $p_{2,1}$ plays a central role in determining the variance. To see this, we emphasize some important points in the proof. By Theorem 2, we have that $Var[BOLSH_{\theta_{a,b},N,K}] - Var[SRPLSH_{\theta_{a,b},K}] = \frac{K\theta_{a,b}}{\pi}(N-1)(p_{2,1} - \frac{\theta_{a,b}}{\pi})$. Therefore the order of $Var[BOLSH_{\theta_{a,b},N,K}]$ and $Var[SRPLSH_{\theta_{a,b},K}]$ is directly determined by the order of $p_{2,1}$ and $\frac{\theta_{a,b}}{\pi}$. Lemma 6 shows that $p_{2,1} < \frac{\theta_{a,b}}{\pi}$ for any $\theta_{a,b} \in (0, \pi/2]$. Together with Theorem 4, we have that $Var[BOLSH_{\theta_{a,b},N,K}] < Var[SRPLSH_{\theta_{a,b},K}]$ for any $\theta_{a,b} \in (0, \pi)$.

A by-product is Corollary 2, which describes the behavior of $Var[BOLSH_{\theta_{a,b},N,K}]$ as N varies ranging from 1 to d. It suggests that when using BOLSH, it is better to set N as

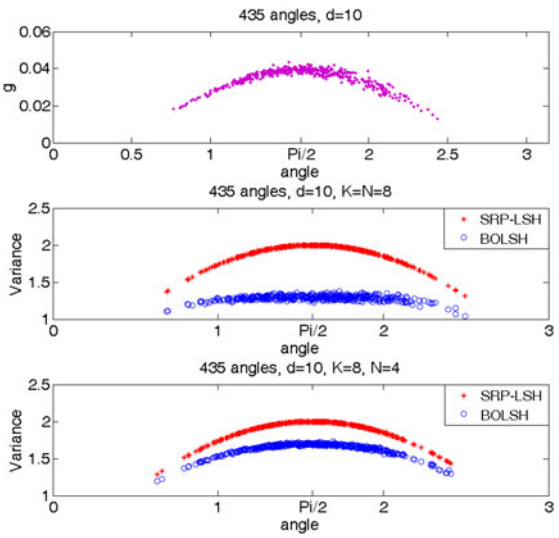


Fig. 2. The value of g (upper), the variances of SRP-LSH and BOLSH against the angle $\theta_{a,b}$ to estimate (middle and bottom).

large as possible. For example, if $K < d$, then we set $N = K$. When $K > d$, for simplicity, assume that $K = L \times d$, then we set $N = d$. As the degree of orthogonality goes up, the variance goes down.

4.3 Numerical Verification

In this section we conduct numerical experiments to verify some of the theoretical results proven in previous sections.

By Corollary 2, we have that $Var[SRPLSH_{\theta_{a,b},K}] - Var[BOLSH_{\theta_{a,b},N,K}] = \frac{K\theta_{a,b}}{\pi}(N-1)(\frac{\theta_{a,b}}{\pi} - p_{2,1}) = \frac{K}{\pi}(N-1)\theta_{a,b}(\frac{\theta_{a,b}}{\pi} - p_{2,1}) = \frac{K}{\pi}(N-1)g(\theta_{a,b})$. Here we define

Definition 5. $g(\theta_{a,b}) = \theta_{a,b}(\frac{\theta_{a,b}}{\pi} - p_{2,1})$.

$g(\cdot)$ is a complicated function of $\theta_{a,b}$, because $p_{2,1}$ is a complicated function of $\theta_{a,b}$. When fixing K and N, the gap between $Var[BOLSH_{\theta_{a,b},N,K}]$ and $Var[SRPLSH_{\theta_{a,b},K}]$ changes linearly with $g(\cdot)$. Thus the value of $g(\cdot)$ directly determines the reduction of variance. The larger the value of $g(\cdot)$, the larger the reduction of $Var[BOLSH_{\theta_{a,b},N,K}]$ over $Var[SRPLSH_{\theta_{a,b},K}]$.

In this section we conduct a numerical verification to depict the behavior of $g(\cdot)$, $Var[SRPLSH_{\theta_{a,b},K}]$ and $Var[BOLSH_{\theta_{a,b},N,K}]$, as $\theta_{a,b}$ changes.

The variance of SRP-LSH is shown as (3). For each angle, we can directly compute the exact value.

By Theorem 2, the variance of BOLSH depends on $p_{2,1}$, so does $g(\cdot)$ (see Definition 5). Since $p_{2,1}$ is a function of $\theta_{a,b}$ in the form of a complicated integral, we use sampling scheme to approximate the value of $p_{2,1}$ for each $\theta_{a,b}$.

We randomly generate 30 points in $\mathbb{R}^{10}$, which involve 435 angles. For each angle, we numerically approximate $p_{2,1}$ using sampling method, where the sample number is 1,000. We set $K = N = 8$ and $K = 8, N = 4$ respectively, and plot the values of $g(\cdot)$ (Definition 5), the variances $Var[SRPLSH_{\theta_{a,b},K}]$ and $Var[BOLSH_{\theta_{a,b},N,K}]$ against various angles $\theta_{a,b}$.

Fig. 2 shows that for $\theta_{a,b} \in (0, \pi)$, BOLSH has a smaller variance than SRP-LSH, which verifies Corollary 3 to some extent. And the reduction of variance is much larger when $N = 8$. Furthermore, the values of $Var[BOLSH_{\theta_{a,b},N,K}]$ and $g(\cdot)$ against $\theta_{a,b}$ are both symmetric about $\theta_{a,b} = \pi/2$, which verifies Theorem 4. And the closer is $\theta_{a,b}$ to $\pi/2$, the larger is the gap between $Var[SRPLSH_{\theta_{a,b},K}]$ and $Var[BOLSH_{\theta_{a,b},N,K}]$.

5 Experiments

We conduct two sets of experiments, angular similarity estimation and approximate nearest neighbor retrieval, to evaluate the effectiveness of the proposed BOLSH method. In the first set of experiments we directly measure the accuracy of estimating pairwise angles, and thus verify the theoretical results developed in previous sections. The second set of experiments then test the performance of BOLSH in approximate nearest neighbor search application.

5.1 Angular Similarity Estimation

In this set of experiments, we evaluate the accuracy of estimating pairwise angular similarity on several data sets. Specifically, we test the effect to the estimation accuracy when the batch size N varies and the code length K is fixed, and vice versa. For each preprocessed data set D, we get D_{LSH} after performing SRP-LSH, and get D_{BOLSH} after performing the proposed BOLSH. We compute the angle between each pair of samples in D, the corresponding Hamming distances in D_{LSH} and D_{BOLSH}. Then we compute the mean squared error between the true angle and the approximated angles from D_{LSH} and D_{BOLSH} respectively. Note that after computing the Hamming distance, we divide the result by $C = K/\pi$ to get the approximated angle.

5.1.1 Data Sets and Preprocessing

We conduct the experiment on the following data sets:

1) *Photo Tourism Notre Dame* patch data set[1] [49], which contains 104,106 patches, each of which is represented by a 128D SIFT descriptor (Photo Tourism SIFT); 2) *SIFT-1M* [10], which contains 1 million 128D SIFT descriptors; 3) *MIR-Flickr*,[2] which contains 25,000 images, each of which is represented by a 3,125D bag-of-SIFT-feature histogram; and 4) *KOS blog entries* from the UCI bag-of-words database, which contains 3,430 documents, represented by 6,906D bag-of-words histogram.

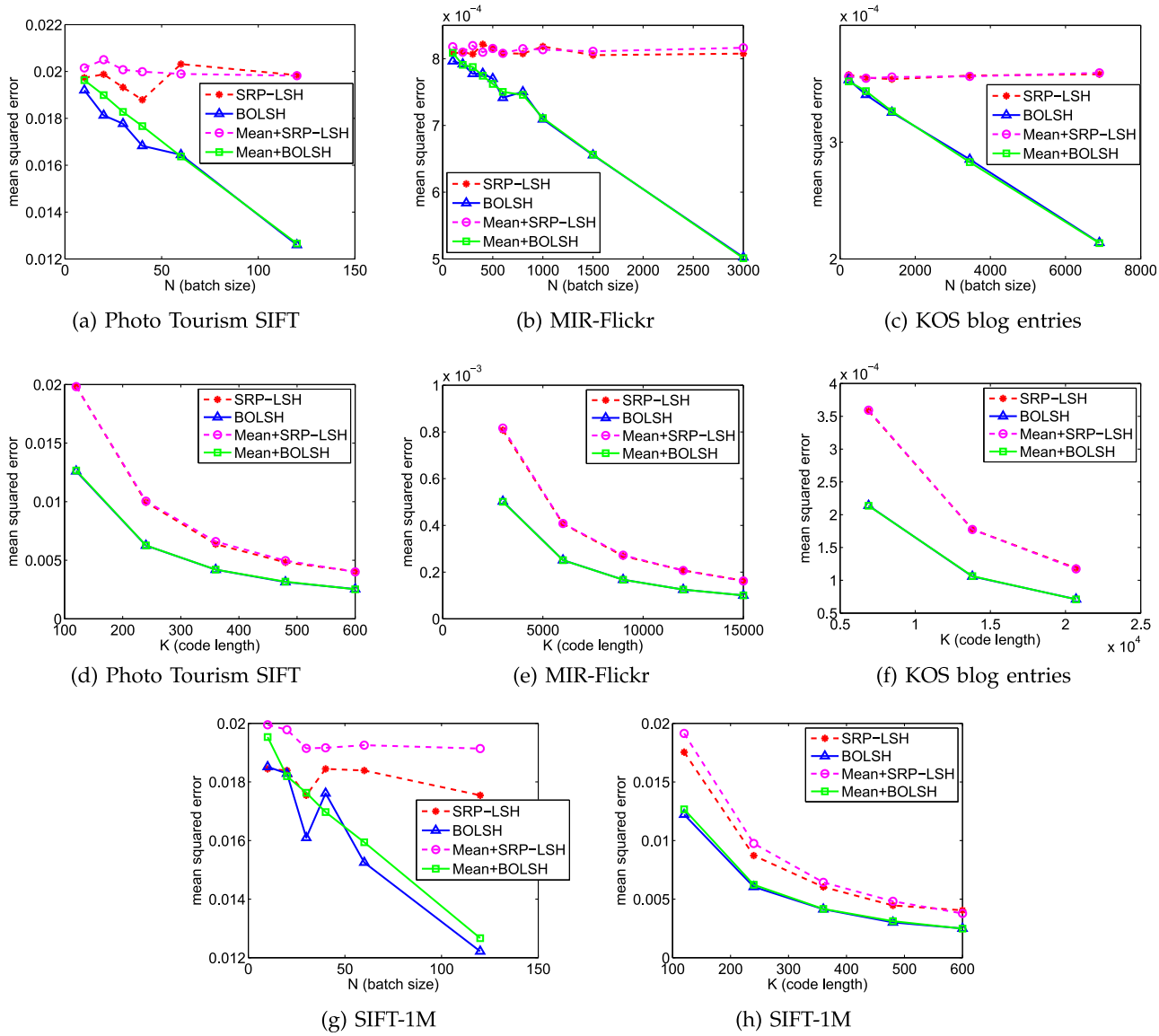
For each data set, we further conduct a simple preprocessing step as in [47], namely, mean-centering each data sample, so as to obtain additional mean-centered versions of the above data sets, Photo Tourism SIFT (mean), MIR-Flickr (mean), and so on.

The experiments on these mean-centered data sets will test the performance of BOLSH when the angles of data pairs are not constrained in $(0, \pi/2]$, but in the whole $(0, \pi)$.

5.1.2 The Effect of Batch Size N and Code Length K

For Photo Tourism SIFT, SIFT-1M and MIR-Flickr, for each (N, K) pair, i.e., batch size N and code length K, we randomly sample 10,000 data, which involve about 50,000,000

1. http://phototour.cs.washington.edu/patches/default.htm.
2. http://users.ecs.soton.ac.uk/jsh2/mirflickr/.



(a) Photo Tourism SIFT (b) MIR-Flickr (c) KOS blog entries

(d) Photo Tourism SIFT (e) MIR-Flickr (f) KOS blog entries

(g) SIFT-1M (h) SIFT-1M

Fig. 3. The effect of batch size N ($1 < N \leq \min(d, K)$) with fixed code length K ($K = N \times L$), and the effect of code length K with fixed batch size N.

data pairs, for KOS blog entries, we randomly sample 300 data. We randomly generate SRP-LSH functions, together with BOLSH functions by orthogonalizing the generated SRP in batches. We repeat the test for 10 times, and compute the mean squared error of the estimation.

To test the effect of batch size N, we fix $K = 120$ for Photo Tourism SIFT and SIFT-1M, $K = 3{,}000$ for MIR-Flickr, and $K = 6{,}900$ for KOS blog entries. To test the effect of code length K, we fix $N = 120$ for Photo Tourism SIFT and SIFT-1M, $N = 3{,}000$ for MIR-Flickr, and $N = 6{,}900$ for KOS blog entries. We repeat the experiment on the mean-centered versions of these data sets, and denote the methods by Mean+SRP-LSH and Mean+BOLSH respectively.

Fig. 3 shows that when using fixed code length K, as the batch size N gets larger ($1 < N \leq \min(d, K)$), the MSE of BOLSH gets smaller, and the gap between BOLSH and SRP-LSH gets larger. Particularly, when $N = K$ and K is close to d, over 30 percent MSE reduction can be observed on all the data sets. This verifies Corollary 2 that the closer the batch size N is to the data dimension d, the larger the variance reduction BOLSH achieves over SRP-LSH. Thus when applying BOLSH, the best strategy would be to set the batch size N as large as possible, i.e., $\min(d, K)$. An informal explanation to this interesting phenomenon is that as the degree of orthogonality of the random projections gets higher, the codes become more and more informative, and thus provide better estimates.

On the other hand, it can be observed that the performances on the mean-centered data sets are similar as those on the original data sets. This shows that when the angle between each data pair is not constrained in $(0, \pi/2]$, but in the whole $(0, \pi)$, BOLSH still gives much more accurate estimations. This verifies to some extent that BOLSH has a smaller variance in the whole $(0, \pi)$.

Fig. 3 also shows that with fixed batch size N BOLSH significantly outperforms SRP-LSH. When increasing the code length K, the accuracies of BOLSH and SRP-LSH shall both increase. The performances on the mean-centered data sets are similar as those on the original data sets.

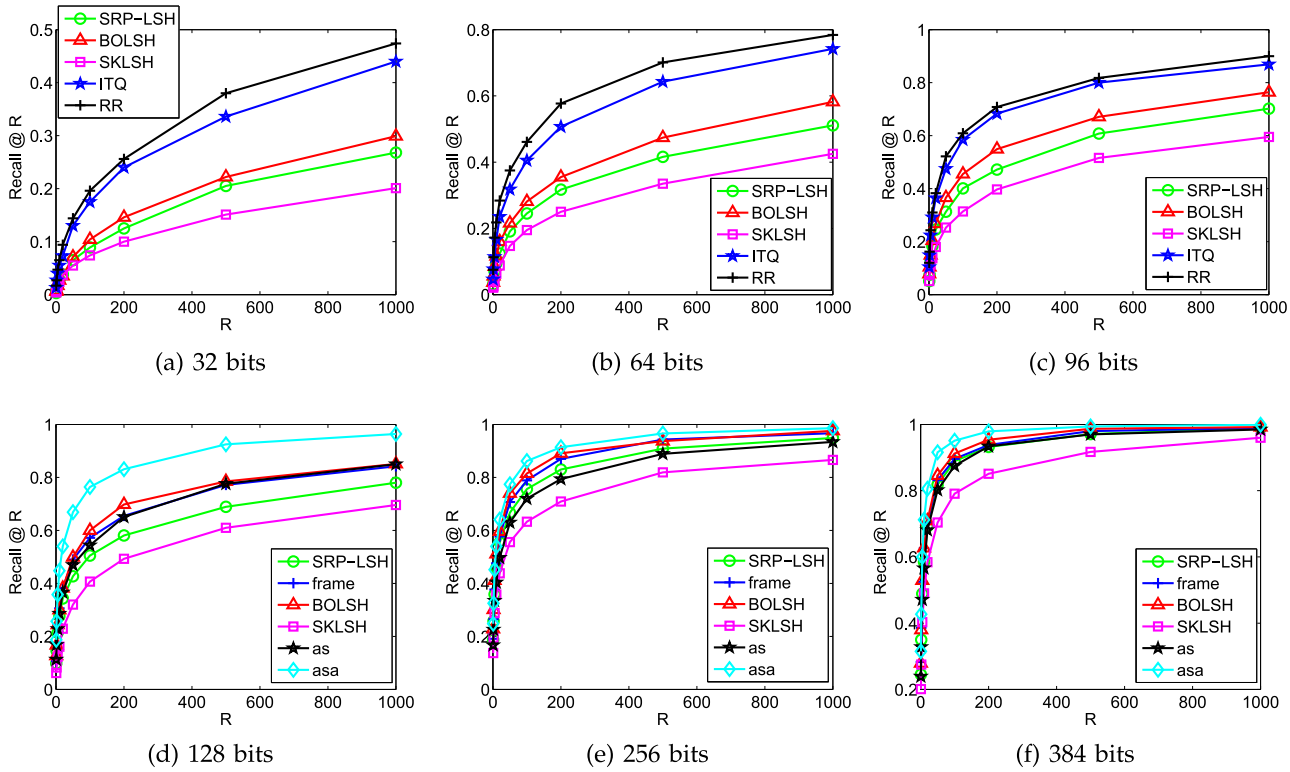


Fig. 4. Approximate nearest neighbor search performance on SIFT-1M for various code lengths.

5.2 Approximate Nearest Neighbor Search

5.2.1 *Approximate Nearest Neighbor Search with Various Code Lengths*

In this section, we conduct approximate nearest neighbor search experiments, and compare BOLSH with several other widely used data-independent and data-dependent binary hashing methods: SRP-LSH, SKLSH [27], LSH+frame (denoted by "frame") [32], Iterative quantization (ITQ) [31], PCA+random rotation (RR) [31], [50], and anti-sparse coding (binary version, denoted by "as") together with its asymmetry version (denoted by "asa") [32]. We can compare these methods because when data are normalized to unit norm, angular similarity changes monotonically with euclidean distance.

Data Set. We use *SIFT-1M* [10] data set as described in Section 5.1.1. We do not use Photo Tourism, MIR-Flickr and KOS blog entries because they are not large enough for retrieval. For SIFT-1M, we randomly sample 1,000 data in query set as queries, while the corpus contains 1 million data, and there is an additional learning set of 100,000 data for learning-based methods. All SIFT descriptors are normalized to unit norm.

Criterion. We adopt recall@R introduced in [10] as our evaluation criterion. Recall@R is the proportion of queries for each of which the true nearest neighbor is ranked within the first R positions.

Experiment setup. For each method to test, if it is a learning based hashing method (ITQ, RR), we first conduct a learning phase using the learning set of the data. Then, we use the method to encode all the data in the database into binary codes. In test phase, each query is encoded into binary code, and then we compute the Hamming distance to the binary code of each data sample in the database, and retrieve the first R candidates with the smallest Hamming distances. Finally we evaluate the performance using recall@R. We repeat the experiment for different code lengths, ranging from small code length of 32 bits, to long code length that exceeds the data dimension. ITQ, RR only generate small codes, while LSH+frame and anti-sparse coding only produce long codes. So we compare them with BOLSH and SRP-LSH in small code length and long code length respectively. For BOLSH, we set $N = \min(K, d)$.

Fig. 4 shows that in general BOLSH performs best among all the data-independent methods compared, and always significantly outperforms SRP-LSH and SKLSH. In particular, when $K \geq d$, except from the asymmetry version of the anti-sparse coding method, BOLSH performs the best among all the methods compared, including the binary version of anti-sparse coding. The asymmetry version achieves the best result when $K \geq d$ since it does not quantize a query to binary code, leaving it a real vector after the coding. And it does not search with Hamming distance but with more computationally expensive dot-product. Therefore in some sense it is not a completely fair comparison between this asymmetry version and the other methods. Besides, the coding time of anti-sparse coding is several magnitudes more than BOLSH and the other methods.

Fig. 4 also shows that in small code length, ITQ and PCA +random rotation significantly outperform the other data-independent methods. But surprisingly, PCA+random rotation outperforms ITQ, which is opposite from the result reported in [31]. As the code length K gets larger, data-independent methods progressively achieve better results, approaching the best performance of the data-dependent

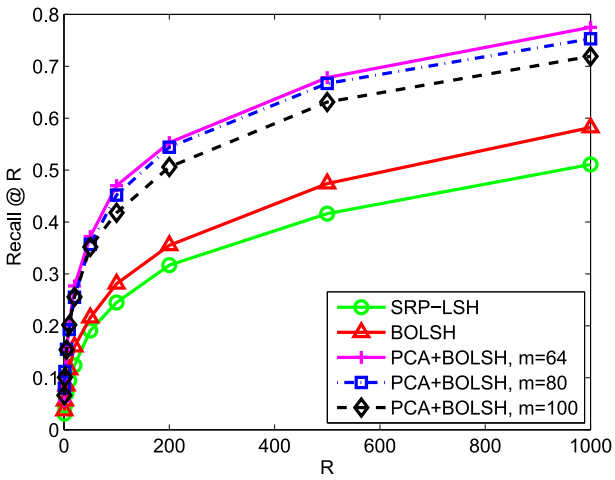


Fig. 5. The effect of dimensionality reduction on the performance of approximate nearest neighbor search on SIFT-1M.

ones. This is the result of the asymptotic convergence guarantees for the data-independent methods. Besides, LSH +frame always performs better than SRP-LSH, as observed in [32]. Note that in the experiment we do not test the product quantization method [10], which is reported to achieve superior results in terms of recall@R on SIFT-1M to some state-of-the-art hashing methods for the same number of bits. However, product quantization is a lookup-based method, different from the Hamming-based binary hashing methods discussed in this paper. As discussed in the Introduction, Hamming-based hashing methods have their own advantages in terms of speed and space.

5.2.2 The Effect of Dimensionality Reduction

In this section we test the effect of dimensionality reduction on the proposed BOLSH method in approximate nearest neighbor search, for $K < d$. The motivation is that, when data is of low intrinsic dimension, conducting a standard dimensionality reduction such as PCA as a preprocessing step may improve the performance.

We first conduct a dimensionality reduction on the data via PCA, to reduce the original dimension d to a middle dimension m. Then we apply BOLSH to further reduce the dimension to K, resulting in K-bit binary codes. We denote the method by PCA+BOLSH. Note that when $m = d$, PCA +BOLSH is the same as BOLSH, and when $m = K = N$, PCA+BOLSH is the same as PCA+random rotation. We conduct the approximate nearest neighbor search experiment with the same setup as described above, on SIFT-1M data set, for $K = N = 64$ and $m = 100, 80, 64$.

Fig. 5 shows that PCA+BOLSH achieves better search results than BOLSH and SRP-LSH. This reveals that SIFT-1M data set has a low intrinsic dimension, and PCA as a preprocessing step does help improve the performance in this case.

6 Conclusions

The proposed BOLSH is a data-independent hashing method which significantly outperforms SRP-LSH. Instead of independent random projections, BOLSH uses batch-orthogonalized random projections. We provide theoretical justifications that BOLSH provides an unbiased estimate of angular similarity, and it has a smaller variance than SRP-LSH, for any angle in $(0, \pi)$. We also provide a lower bound on the variance reduction. Experiments show that with the same length of binary code, BOLSH achieves significant mean squared error reduction over SRP-LSH in estimating angular similarity. And BOLSH also performs best among several widely used data-independent hashing methods in approximate nearest neighbor search experiments. The BOLSH method is closely related to many recently proposed PCA-style learning-based hashing methods, which learn orthogonal projections. Although BOLSH is purely probabilistic and data-independent, the model of orthogonal random projection together with its theoretical justifications can help gain more insights and a better understanding of these learning-based hashing methods. The algorithm is simple, easy to implement and it can be integrated as a basic module in more complicated algorithms. Its potential applications include approximate nearest neighbor search, clustering, near-duplicate detection, and others that require massive angle-related computations.

Acknowledgments

This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403 and 2012CB316301), the National Natural Science Foundation of China (Grant No. 61332007, 61273023 and 91120011), the Beijing Natural Science Foundation (Grant No. 4132046), and the Tsinghua University Initiative Scientific Research Program (Grant No. 20121088071). This research is partially supported by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office. This work was supported in part to Dr. Qi Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, and 2012 UTSA START-R Research Award respectively. This work was supported in part by National Science Foundation of China (NSFC) 61128007. Part of the work appeared in NIPS 2012.

References

[1] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[2] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. ACM Symp. Theory Comput.*, 1998, pp. 604–613.

[3] D. Ravichandran, P. Pantel, and E. H. Hovy, "Randomized algorithms and NLP: Using locality sensitive hash functions for high speed noun clustering," in *Proc. 43rd Annu. Meet. Assoc. Comput. Linguistics*, 2005, pp. 622–629.

[4] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," *Comput. Netw.*, vol. 29, nos. 8-13, pp. 1157–1166, 1997.

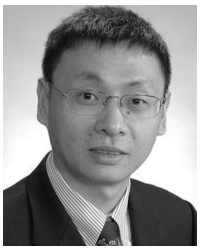
[5] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and TF-IDF weighting," in *Proc. Brit. Mach. Vis. Conf.*, 2008, vol. 810, pp. 812–815.

[6] A. Broder, "On the resemblance and containment of documents," in *Proc. Compression Complexity of Sequences*, 1997, pp. 21–29.

[7] G. S. Manku, A. Jain, and A. D. Sarma, "Detecting near-duplicates for web crawling," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 141–150.

[8] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 17–24.
[9] 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.
[10] H. Jégou, 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.
[11] P. Li, A. Shrivastava, J. L. Moore, and A. C. König, "Hashing algorithms for large-scale learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2672–2680.
[12] M. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. ACM Symp. Theory Comput.*, 2002, pp. 380–388.
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[14] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[15] P. Jain, S. Vijayanarasimhan, and K. Grauman, "Hashing hyperplane queries to near points with applications to large-scale active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 928–936.
[16] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2130–2137.
[17] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian, "Super-bit locality-sensitive hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 108–116.
[18] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013.
[19] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. Symp. Comput. Geom.*, 2004, pp. 253–262.
[20] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2003, pp. 76–85.
[21] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Multimedia*, 2004, pp. 869–876.
[22] Y.-H. Kuo, K.-T. Chen, C.-H. Chiang, and W. H. Hsu, "Query expansion for hash-based image object retrieval," in *Proc. ACM Multimedia*, 2009, pp. 65–74.
[23] B. Matei, Y. Shan, H. S. Sawhney, Y. Tan, R. Kumar, D. F. Huber, and M. Hebert, "Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1111–1126, Jul. 2006.
[24] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 456–463.
[25] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 459–468.
[26] P. Li and A. C. König, "b-bit minwise hashing," in *Proc. Int. World Wide Web Conf.*, 2010, pp. 671–680.
[27] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.
[28] P. Li, T. Hastie, and K. W. Church, "Very sparse random projections," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 287–296.
[29] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.
[30] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learning*, 2011, pp. 1–8.
[31] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 817–824.
[32] H. Jégou, T. Furon, and J.-J. Fuchs, "Anti-sparse coding for approximate nearest neighbor search," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2012, pp. 2029–2032.
[33] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2938–2945.
[34] Y. Mu, J. Shen, and S. Yan, "Weakly-supervised hashing in kernel space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3344–3351.
[35] J. Wang, S. Kumar, and S.-F. Chang, "Sequential projection learning for hashing with compact codes," in *Proc. Int. Conf. Mach. Learning*, 2010, pp. 1127–1134.
[36] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," in *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.
[37] 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.
[38] J. Wang, O. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3424–3431.
[39] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. Mach. Learning*, 2011, pp. 353–360.
[40] C. Strecha, A. A. 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.
[41] R. Salakhutdinov and G. E. Hinton, "Semantic hashing," *Int. J. Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.
[42] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: efficient indexing for high-dimensional similarity search," in *Proc. 33rd Int. Conf. Very Large Data Bases*, 2007, pp. 950–961.
[43] V. Satuluri and S. Parthasarathy, "Bayesian locality sensitive hashing for fast similarity search," in *Proc. VLDB Endowment*, vol. 5, no. 5, pp. 430–441, 2012.
[44] M. Norouzi, A. Punjani, and D. J. Fleet, "Fast search in hamming space with multi-index hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3108–3115.
[45] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis.: Part I*, 2008, pp. 304–317.
[46] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
[47] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.
[48] Y. Chikuse, *Statistics on Special Manifolds*. New York, NY, USA: Springer, Feb. 2003.
[49] S. A. J. Winder and M. Brown, "Learning local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[50] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

Jianqiu Ji received the BE degree from the Department of Computer Science and Technology, Tsinghua University, in 2010. Currently, he is working toward the PhD degree in the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University. His supervisor is Prof. Bo Zhang. His current research interest includes hashing method.



Shuicheng Yan is currently an associate professor in the Department of Electrical and Computer Engineering at the National University of Singapore, and the founding lead of the Learning and Vision Research Group. His research areas include computer vision, multimedia, and machine learning, and he has authored/coauthored more than 340 technical papers over a wide range of research topics, with Google Scholar citation $>$9,900 times and H-index-43. He is an associate editor of the *IEEE Transactions on Circuits and Systems for Video Technology* and the *ACM Transactions on Intelligent Systems and Technology*, and has been serving as the guest editor of the special issues for the *IEEE Transactions on Multimedia* and *Computer Vision and Image Understanding*. He received the Best Paper Awards from ACM MM'12 (demo), PCM'11, ACM MM'10, ICME'10, and ICIMCS'09, the winner prizes of the classification task in PASCAL VOC 2010-2012, the winner prize of the segmentation task in PASCAL VOC 2012, the honourable mention prize of the detection task in PASCAL VOC'10, 2010 TCSVT Best Associate Editor (BAE) Award, 2010 Young Faculty Research Award, 2011 Singapore Young Scientist Award, and 2012 NUS Young Researcher Award. He is a senior member of the IEEE.

Jianmin Li received the PhD degree in computer application from the Department of Computer Science and Technology, Tsinghua University, in 2003. Currently, he is an associate professor in the Department of Computer Science and Technology, Tsinghua University. His main research interests include image and video analysis, image and video retrieval, and machine learning. He has published more than 50 journal and conference papers. He received the second class Technology Innovation Award by State Administration of Radio Film and Television in 2009.

Guangyu Gao received the MS degree in computer science and technology from Zhengzhou University, China, in 2007 and the PhD degree in computer science and technology from the Beijing University of Posts and Telecommunications (BUPT) in 2013. He is an assistant professor at the School of Software, Beijing Institute of Technology, China. He also spent about 1 year at the National University of Singapore as a government-sponsored Joint-PhD student from July 2012 to April 2013. His current research interests include applications of multimedia, computer vision, video analysis, machine learning, and Internet of Things (IoT).

Qi Tian (M'96-SM'03) received the BE degree in electronic engineering from Tsinghua University, China, in 1992, the MS degree in electrical and computer engineering from Drexel University in 1996, and the PhD degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign in 2002. He is currently a professor in the Department of Computer Science at the University of Texas at San Antonio (UTSA). He took a one-year faculty leave at Microsoft Research Asia (MSRA) during 2008-2009. His research interests include multimedia information retrieval and computer vision. He has published more than 230 refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA and he also received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He received the Best Paper Awards in PCM 2013, MMM 2013, and ICIMCS 2012, the Top 10 percent Paper Award in MMSP 2011, the Best Student Paper in ICASSP 2006, and the Best Paper Candidate in PCM 2007. He received 2010 ACM Service Award. He is currently the Associate Editor of *IEEE Transactions on Multimedia (TMM)*. He is the guest editors of the *IEEE Transactions on Multimedia, Journal of Computer Vision and Image Understanding, Pattern Recognition Letter, EURASIP Journal on Advances in Signal Processing, Journal of Visual Communication and Image Representation*, and is in the editorial board of the *IEEE Transactions on Circuit and Systems for Video Technology*, *Multimedia Systems Journal, Journal of Multimedia* and the *Journal of Machine Visions and Applications*. He is a senior member of the IEEE.

Bo Zhang received the graduate degree from the Department of Automatic Control, Tsinghua University, in 1958. He is currently a professor of Computer Science and Technology Department, Tsinghua University, Beijing, China. His main research interests include artificial intelligence, robotics, intelligent control, and pattern recognition. He has published about 150 papers and three monographs in these fields. He is a fellow of the Chinese Academy of Sciences.