Learning Hash Functions Using Sparse Reconstruction

Yong Yuan, Xiaoqiang Lu, Xuelong Li
Center for OPTical IMagery Analysis and Learning (OPTIMAL),
State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics,
Chinese Academy of Sciences, Xi'an 710119, Shaanxi, P. R. China.
yuanyong@opt.cn, {luxiaoqiang, xuelong_li}@opt.ac.cn

ABSTRACT

Approximate nearest neighbor (ANN) search is becoming an increasingly important technique in large-scale problems. Recently many approaches have been developed due to fast query and low storage cost. Although most of them have realized the importance of the data structure, they neglected the sparse relationship of the data. To build a balance between the adjusted covariance matrix and the minimum reconstruction error of data points, this paper proposes a novel method based on sparse reconstruction to learn more compact binary codes under $l_{2,1}$-norm constraint. Experiments demonstrate that the proposed method, named as sparse reconstruction hashing, outperforms several other state-of-the-art methods when tested on a few benchmark datasets.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Experimentation, Performance

Keywords

Image search, ANN, hashing, sparse reconstruction

1. INTRODUCTION

Nearest neighbor (NN) search has been widely applied to large-scale vision problems including image retrieval, object recognition and many other computer vision problems. Recently, with the explosive growth of data, many research efforts have been devoted to investigating the alternative solution-ANN search, which allows fast search in large database. To address this problem, most of tree-based methods have been proposed. However, for high-dimensional data, the performance of tree-based methods suffer significantly with their performance drastically degrading to exhaustive linear search. Besides, since the data structure based on tree is bigger than the original data itself, it also suffers from memory constraints.

To overcome these issues, many researchers have proposed to use hashing technique for ANN search. There is a general consensus among these researchers that semantically similar data points should be mapped to the same or close hash buckets in ANN search. One of the greatest advantage of hashing-based techniques is that the query time is constant or sub-linear. Moreover, the storage reduces substantially as they learn compact codes representation for the data while preserving the similarity structure of the original feature space. Hence, hashing-based methods can achieve fast query time with low storage requirement, which provide a popular candidate for efficient ANN search in large-scale datasets.

Currently, a lot of hashing-based methods have been proposed to design effective compact hash functions. According to whether they make use of dataset to learn the hash functions, hashing-based methods can be roughly divided into two categories, i.e. data-independent hashing methods and data-dependent hashing methods.

Data-independent hashing methods usually construct a set of hash functions using randomization without training stage. The representative data-independent methods include Local Sensitive Hashing (LSH) [1], its extension and Shift-Invariant Kernel-based Hashing (SIKH) [2]. Due to the data-blindness, data-independent hashing methods usually need a large number of bits and neglect data internal structure. Besides, longer codes decrease greatly the collision probability between close data samples in the original space. Hence, multiple hash tables are introduced to enlarge the probability to achieve reasonable recall, but this results in increase of query time and big storage requirement.

In general, data-dependent hashing methods are shown to be superior to data-independent hashing methods. Unlike data-independent hashing methods, data-dependent hashing methods can make use of learning techniques to construct a set of hash functions. According to whether they make use of the label information or not, the data-dependent hashing methods can be summarized into three categories, namely supervised, semi-supervised [3] and unsupervised hashing methods.

Supervised hashing methods try to take advantage of labeled training samples to improve hashing performance such as Semantic Hashing [4], Binary Reconstructive Embedding (BRE) [5], and Kernel-based Supervised Hashing (KSH) [6]. However, the main problem with the existing supervised hashing techniques is that they need many labeled samples, which is a tough task to label when the database is large.

ICIMCS'14, July 10–12, 2014, Xiamen, Fujian, China.
Copyright 2014 ACM 978-1-4503-2810-4/14/07 ...$15.00.

To handle the aforementioned problem, unsupervised hashing methods, including Kernelized Locality-Sensitive Hashing (KLSH) [7], Spectral Hashing (SH) [8], Multidimensional Spectral Hashing (MDSH) [9], PCA-ITQ [10], and K-Means Hashing (KMH) [11], can achieve binary codes from given data samples by unsupervised learning. Since unsupervised methods don't need labeled data, they can be widely applied to different data domains.

In this paper, we focus on unsupervised method and propose a novel method using sparse reconstruction to learn hash functions. First, l_{21}-norm is introduced into *sparsity preserving projections* (SPP) framework to preserve potential discrimination information of hash functions. Second, we construct an objective function to build a balance between the adjusted covariance matrix and the minimum reconstruction error of data points. The main contributions of this paper are briefly outlined as follows:

- It has been pointed out that SPP contains natural discriminating information. To make full use of the advantage, a novel sparse reconstruction method is proposed to learn hash functions. Furthermore, the proposed method uses a l_{21}-norm instead of l_1-norm to capture some intrinsic geometric information of hash functions.
- The proposed method takes both information entropy maximum and data structure into consideration simultaneously. To the best of our knowledge, most of existing hashing methods treat them as two independent phases, but actually they may be inter-related. The proposed method considers them simultaneously and we present an efficient algorithm to solve the formulation.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 presents the details of our method including optimization method. Section 4 gives comparison results to evaluate the effectiveness of the proposed method. Section 5 concludes the paper.

2. RELATED WORK

Since some annotations will be used later, we first introduce them for convenience. Suppose the dataset $X \in R^{d\times N}$ consists of N data points $\{x_i\}_{i=1}^N$, $x_i \in R^d$. The purpose of hashing-based method tries to learn a set of hash functions $h_k(x) \in H$ $(k = 1, 2, \cdots, K)$ to map each data point x_i to a K-bit low-dimensional hash code $H(x_i) = [h_1(x_i), h_2(x_i), \cdots, h_K(x_i)]$, where $h_k(x_i)$ belongs to {-1,1}. Assuming the data points are zero-centered, for each bit $k = 1, \cdots, K$, the binary encoding function is defined as $h_k(x_i) = sgn(w_k^T x_i)$, where w_k is a column vector of hyperplane coefficient. The function $sign(v)$ is defined as 1 if input variable $v > 0$ and -1 otherwise.

Following the formulation of [8], in order to generate an efficient code, each bit requires a 50% chance of being one or zero. According to information theory's maximum entropy rule, the variance of hash functions are required to be maximized and the following objective function can be obtained:

$$J_1(W) = \sum_{k=1}^{K} var(h_k(x)) = \sum_{k=1}^{K} var(sgn(w_k^T x)). \tag{1}$$

As shown in [10], solving the objective function in Eq.1 is intractable since the requirement of exact balancedness makes the above objective function intractable. By relaxing the signed magnitude as shown in [3], the objective function (1) can be transformed as follows:

$$\begin{aligned}\tilde{J}_1(W) &= \sum_{k=1}^{K} E(||w_k^T x||_2^2) = \frac{1}{n}\sum_{k=1}^{K} w_k^T X X^T w_k \\ &= \frac{1}{n} tr(W^T X X^T W).\end{aligned} \tag{2}$$

For a pair of points x_i and x_j, if x_i and x_j are similar to each other, it's important for hash functions to encourage them to be encoded into similar binary codes since the corresponding weight $t_{i,j}$ between them is large. Hence it's reasonable to maximize the following objective function according to [12]:

$$\tilde{J}_2(W) = \sum_{i,j=1}^{N} t_{ij} < H(x_i), H(x_j) >, \tag{3}$$

where the weight element $t_{i,j}$ (T is the corresponding matrix) is denoted the same as $s_{i,j}$ in [12]. By relaxing the $sgn(w^T x)$ to the signed magnitude, the objective function in Eq.3 can be rewritten as:

$$\begin{aligned}\tilde{J}_2(W) &= \sum_{i,j=1}^{N} t_{ij} H(x_i)^T H(x_j) = \sum_{i,j=1}^{N} t_{ij} (W^T x_i)^T W^T x_i \\ &= tr(W^T X T X^T W).\end{aligned} \tag{4}$$

By considering the regularization term in Eq.(2), Eq.(4) can be transformed as follow:

$$\begin{aligned}\tilde{J}_2(W) &= \max_W tr(W^T X T X^T W + \lambda W^T X X^T W) \\ &= \max_W tr(W^T M W),\end{aligned} \tag{5}$$

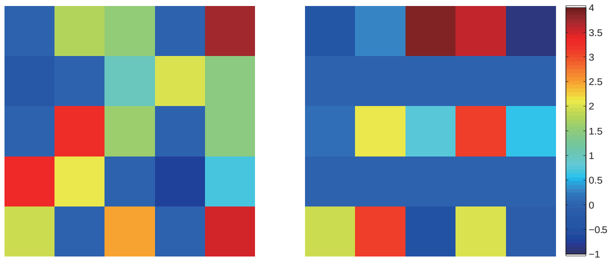
where $M = XTX^T + \lambda XX^T$ is a $d \times d$ matrix, which can be referred as the adjusted covariance matrix [3]. λ is the regularized parameter in Eq.5. The first term maps the similar points in the original space to the same or close buckets. The second term guarantees each bit contains maximum information.

3. THE PROPOSED METHOD

SPP is a popular reconstruction technique, which aims to preserve the sparse reconstructive relationship of the data. The main advantage of SPP is that it contains discriminating information since sparse representation has natural discriminating power. For a given sample x_i, it's expected to sparsely represent each data point from the dataset X. The reconstruction estimation for sparse representation of x_i can be obtained by minimizing the loss function under a penalized constraint:

$$\begin{aligned}&\min_{s_i} ||s_i||_1 \\ &s.t.\ x_i = X s_i, i = 1, 2, ..., N \\ &1 = 1^T s_i,\end{aligned} \tag{6}$$

where $s_i \in R^N$ is a sparse coefficient vector. It's reasonable to treat $S = [s_1, s_2, \cdots, s_N] \in R^{N\times N}$ as the affinity weight matrix since the element s_{ij} in S reflects a close relation between x_i and x_j. The l_1-norm $||\cdot||_1$ guarantees the sparsity of s_i, that is, many elements in vector s_i are zeros. The



(a) l1-norm (b) l21-norm

Figure 1: Illustration for the sparse weight matrix by (a) l_1-norm and (b) $l_{2,1}$-norm

weight matrix S not only can measure the similarity among different samples, but also may captures intrinsic structure information due to the element s_i invariant to rotations and rescalings. Moreover, the nonzero entries in the sparse matrix S may help to distinguish the samples from the given class even if no class-labels provided, thus the sparse reconstructive weight vector tends to contain potential discriminant information.

Ding and zhou (2006) first introduced the $l_{2,1}$-norm of a matrix as a rotational invariant l_1-norm, and it has attracted increasing attention. Fig. 1 illustrates a matrix under l_1-norm constraint compared with $l_{2,1}$-norm. From Fig. 1, the $l_{2,1}$-norm has a structured sparsity attribute compared with l_1-norm. Generally, $l_{2,1}$-norm based priors offer a general way to take the structure into consideration, therefore the $l_{2,1}$-norm constraint on matrix S has potential discrimination.

In this paper, by imposing the $l_{2,1}$-norm on S, the set of hash functions can be obtained $h_k(x) \in H(k = 1, 2, ..., K)$ from the following objective function:

$$\min_{h_k,S} \sum_{k=1}^{K} \sum_{i=1}^{N} \|h_k(x_i) - h_k(Xs_i)\|^2 + \alpha \|S\|_{2,1}, \qquad (7)$$

where α is a balance parameter to adjust the importance of structure sparsity. The first term in Eq.7 tries to reconstruct each data point to obtain the minimal loss of hash functions. The last term $l_{2,1}$-norm of matrix S regularizes all the elements $\{s_i\}_{i=1}^{N}$ corresponding to the training dataset and is defined as $||S||_{2,1} = \sum_{i=1}^{N} ||s_i||$. s_i is denoted as the ith column of S.

As mentioned in the related work, the binary encoding hash functions are defined as $h_k(x_i) = sgn(w_k^T x_i), (k = 1, 2, \cdots, K)$. It's not easy to solve the Eq.7 due to the non-differentiability of $sgn(\cdot)$ function. In this case, we relax $sgn(w_k^T x_i)$ to the signed magnitude $w_k^T x_i$, and the first term in the Eq.7 can be transformed to the following form:

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{i=1}^{N} ||h_k(x_i) - h_k(Xs_i)||^2 = \sum_{k=1}^{K} \sum_{i=1}^{N} ||w_k^T x_i - w_k^T X s_i|| \\ &= \mathrm{tr}\left[W^T (XX^T - XSX^T - XS^T X^T + XSS^T X^T) W\right] \\ &= \mathrm{tr}\left(W^T X H X^T W\right), \end{aligned} \qquad (8)$$

where $V = (I - S - S^T + SS^T)$. Replacing the first term in the Eq.7 with Eq.8, then the Eq.7 can be transformed as follows:

$$\tilde{J}_3(W, S) = \min_{W,S} \mathrm{tr}\left(W^T X V X^T W\right) + \alpha \|S\|_{2,1}. \qquad (9)$$

By constructing an auxiliary function, Eq.9 can be rewritten as:

$$\tilde{J}_3(W, S) = \min_{W,S} \mathrm{tr}\left(W^T X V X^T W\right) + \alpha \mathrm{tr}\left(SUS^T\right), \qquad (10)$$

where $U \in R^{N \times N}$ is a diagonal matrix and its ith element is defined as:

$$U_{ii} = \frac{1}{2\,\|s_{i\cdot}\|_2}. \qquad (11)$$

To build a balance between the adjusted covariance matrix in Eq.5 and the minimum reconstruction error of data points in Eq.10, the final objective function is proposed by minimizing the following formulation:

$$\begin{aligned} \tilde{J}(W, S) &= \min_{W,S} \frac{J_3(W, S)}{J_2(W)} \\ &= \min_{W,S} \frac{\mathrm{tr}\left(W^T X V X^T W\right) + \alpha \mathrm{tr}\left(SUS^T\right)}{\mathrm{tr}(W^T X T X^T W + \lambda W^T X X^T W)}. \end{aligned} \qquad (12)$$

The proposed objective function in Eq.12 can be divided into two alternating steps: fixing the sparse coefficient matrix S to learn the map W and learning W while fixing S. These two steps are described in detail below.

Fix S and update W. When the sparse coefficient matrix S is fixed, Eq.12 can be transformed as the following objective function:

$$\begin{aligned} \tilde{J}(W) &= \min_{W} \frac{\mathrm{tr}\left(W^T X V X^T W\right)}{\mathrm{tr}(W^T X (T + \lambda I) X^T W)} \\ &= \min_{W} \frac{\mathrm{tr}\left(W^T X V X^T W\right)}{\mathrm{tr}(W^T X G X^T W)}, \end{aligned} \qquad (13)$$

where $G = (T + \lambda I)$. The solution of Eq.13 can be obtained by using generalized eigenvalue decomposition problem. The optimal map matrix W can be obtained by picking up the eigenvectors corresponding to the first K (K is the length of hash codes mentioned above) smallest eigenvalues.

Fix W and update S. For a fixed W, Eq.12 can be transformed as follow:

$$\tilde{J}(S) = \min_{S} \mathrm{tr}\left(DVD^T\right) + \alpha \mathrm{tr}\left(SUS^T\right), \qquad (14)$$

where $D = W^T X$. Taking the derivative of Eq.14 with respect to S and setting it to zeros, we can get the sparse coefficient matrix:

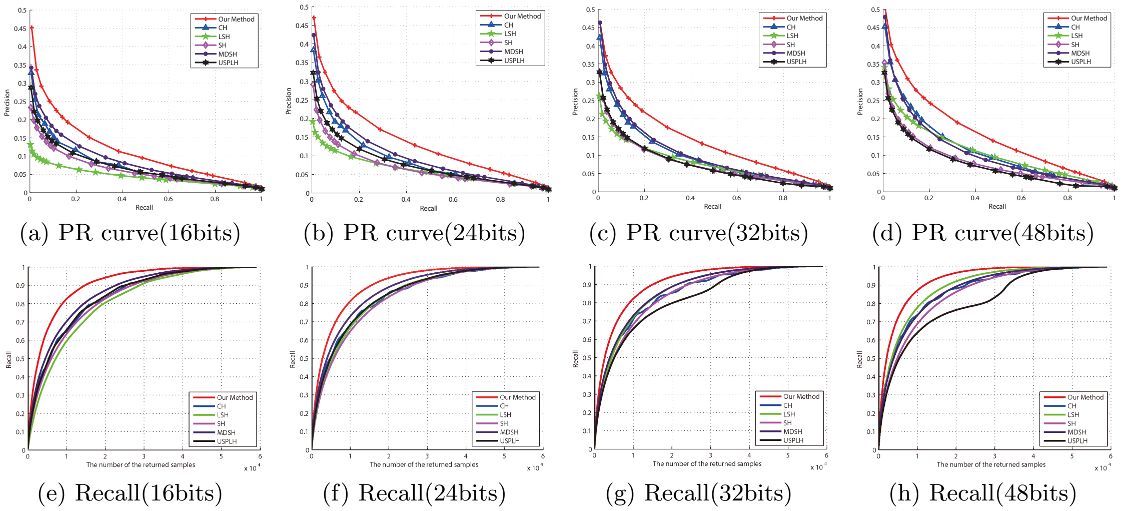
$$S = D^T D (D^T D + \alpha U)^{-1}. \qquad (15)$$

Since $\|s_{i\cdot}\|_2$ may be zero in real world application, the ith element in diagonal matrix U can be redefined as $U_{ii} = \frac{1}{\|s_{i\cdot}\|_2 + \zeta}$ (ζ is a very small constant) in practice.

From the above discussion, the proposed method is summarized in Algorithm 1.

4. EXPERIMENTS

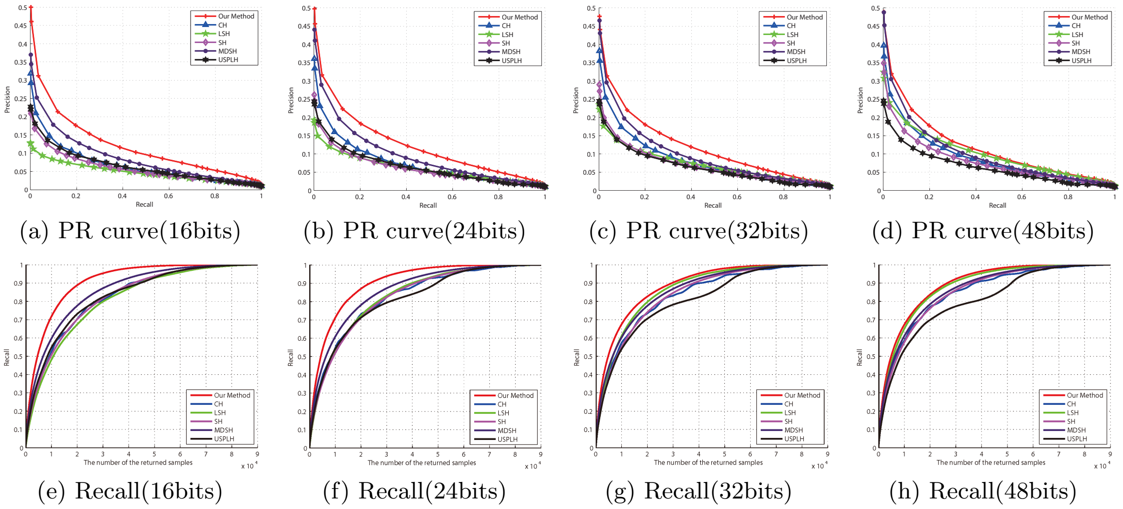
To evaluate the effectiveness of the proposed method, we run large-scale image retrieval on two benchmark datasets, i.e. CIFAR-10 and 100K TinyImage, which have been adopted widely in the evaluation of hashing methods [3, 5, 6, 9, 10]. The first data set CIFAR-10 contains a total of



(a) PR curve(16bits) (b) PR curve(24bits) (c) PR curve(32bits) (d) PR curve(48bits)

(e) Recall(16bits) (f) Recall(24bits) (g) Recall(32bits) (h) Recall(48bits)

Figure 2: Precision-Recall(PR) (a)(b)(c)(d) and Recall (e)(f)(g)(h) curves on CIFAR-10 image dataset.



(a) PR curve(16bits) (b) PR curve(24bits) (c) PR curve(32bits) (d) PR curve(48bits)

(e) Recall(16bits) (f) Recall(24bits) (g) Recall(32bits) (h) Recall(48bits)

Figure 3: Precision-Recall(PR) (a)(b)(c)(d) and Recall (e)(f)(g)(h) curves on 100k Tiny Image dataset.

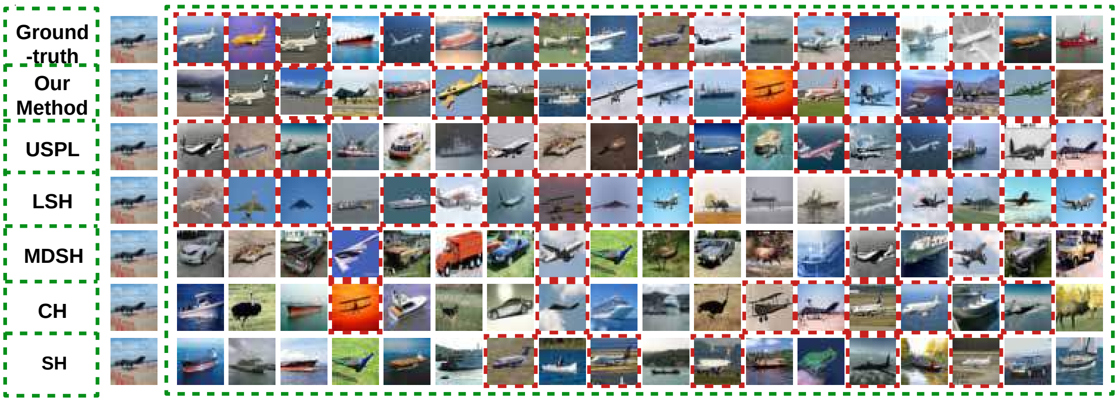


Figure 4: Retrieve an airplane example image using different hash methods

60K with size 32×32 color images from ten class (airplane, automobile, bird, cat, deer, frog, horse, ship and truck). Each class contains 6K samples and the images assigned to a mutually exclusive class label. Every image is represented by a 320-dimensional GIST feature vector. We randomly partitioned it into two parts: a training set of 59K images

Algorithm 1 Sparse Reconstruction Hashing

```
Input: A set of training data $X$ (zero-centered), length of
    hash codes $K$, the parameters $\alpha$, $\eta$.
Output: the sparse weight matrix $S$, the map matrix $W$.
Initialize $S_0 = I_{N \times N}$, and compute the diagonal matrix
    $U$, the $ith$ element of $U_{ii} = \frac{1}{\|s_{i\cdot}\|_2+\varsigma}$.
repeat
    Computer the map matrix $W$ by using generalized
        eigenvalue decomposition problem in Eq.13.
    Computer the sparse matrix $S$ by Eq.15.
    Update the diagonal matrix $U$.
until convergence
```

and a test set of 1K images for query. The second data set is 100K Tiny Images, a subset sampled from the large million Tiny Images. It contains 100K images and each image is presented by a 384-d GIST descriptor. The entire dataset is divided into two parts as well: 90K for training and 10K for test. Since the 100K Tiny Images dataset has no semantic labels, we consider the groundtruth as each test query's n Euclidean nearest neighbors including CIFAR-10. Here n is set to one percent of the size of training data set and the parameters α and η in the object function both set to 0.1.

Since the proposed method is based on unsupervised learning, we compare it with five state-of-the-art unsupervised hashing methods including LSH [1], SH [8], MDSH [9], CH [12], and USPLH [13], and *hamming ranking* search strategy is adopted commonly in hashing methods [3, 8, 10]. *Hamming ranking* sorts all the data points according to their distance to a specific query and the first N samples in the ranked list of the query will be returned. It's an exhaustive linear search method but fast implementation. Besides, it provides better quality measurement of the Hamming embedding though the complexity is linear to the size of the dataset. For a quantitative performance comparison, we adopt precision-recall curses and recall curves of the returned samples number to illustrate the performance of different methods.

The performance comparison on two benchmark datasets are shown in Fig. 2 and Fig. 3, respectively. From the precision-recall curves in Fig. 2 and Fig. 3, it clearly observed that the proposed method obtains larger areas under the precision-recall curves compared with the competing hashing methods. It's not very surprising that the proposed hashing method obtains the best performance since the proposed method keeps a sparse structure under $l_{2,1}$-norm constraint while taking adjusted covariance matrix into consideration. Fig. 2 and Fig. 3 also show a little difference on the two benchmark datasets. On CIFAR-10 dataset, the proposed method's advantage is very obvious compared with others methods as the encoding length increases. Meanwhile, On 100K Tiny Images dataset, when the encoding length increases to 48 bits, though the proposed method still performs best, the advantage is no longer obvious. The main reason is that the training set may not be enough while the test query is a little large. Furthermore, the recall curves in Fig. 2 and Fig. 3 also illustrate that the proposed method can obtain higher recall rate for the returned samples number on the two benchmark datasets.

Finally, to obtain a qualitative visual result, Fig. 4 shows the retrieved result of the proposed method and five competing methods on a airplane example image.

5. CONCLUSIONS

In this paper, a novel sparse reconstruction is proposed by introducing the $l_{2,1}$-norm into SSP framework to preserve potential discrimination information of hash functions. Moreover, we construct an objective function to build a balance between the adjusted covariance matrix and the minimum reconstruction error of data points. The experimental results on two large-scale benchmark datasets show the benefits of our method. In future work, more efficient structure models and labels information will be explored to solve ANN search problem.

6. ACKNOWLEDGMENTS

This project is supported by the National Basic Research Program of China (973 Program) (Grant No. 2012CB719905), the National Natural Science Foundation of China (Grant Nos: 61125106, 91120302, and 61100079).

7. REFERENCES

[1] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999.

[2] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*, pages 1509–1517, 2009.

[3] J. Wang, O. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.

[4] R. Salakhutdinov and G. E. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009.

[5] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, pages 1042–1050, 2009.

[6] W. Liu, J. Wang, R. Ji, Y. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081, 2012.

[7] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, pages 2130–2137, 2009.

[8] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008.

[9] Y. Weiss, R. Fergus, and A. Torralba. Multidimensional spectral hashing. In *ECCV (5)*, pages 340–353, 2012.

[10] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. on PAMI*, 35(12):2916–2929, 2013.

[11] K. He, F. Wen, and J. Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *CVPR*, pages 2938–2945, 2013.

[12] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu. Complementary hashing for approximate nearest neighbor search. In *ICCV*, pages 1631–1638, 2011.

[13] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *ICML*, pages 1127–1134, 2010.