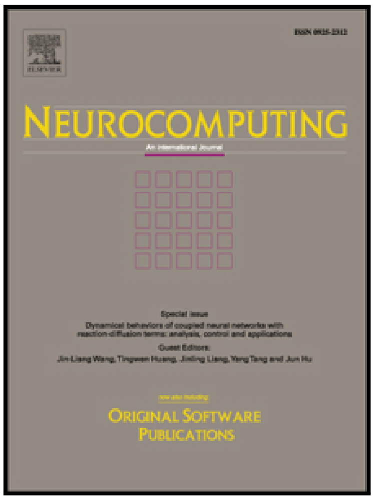
Accepted Manuscript



Large-Scale Image Retrieval with Sparse Embedded Hashing

Guiguang Ding, Jile Zhou, Yuchen Guo, Zijia Lin, Sicheng Zhao, Jungong Han

PII: S0925-2312(17)30160-1
DOI: 10.1016/j.neucom.2017.01.055
Reference: NEUCOM 17972

To appear in: *Neurocomputing*

Received date: 3 July 2016
Revised date: 5 January 2017
Accepted date: 8 January 2017

Please cite this article as: Guiguang Ding, Jile Zhou, Yuchen Guo, Zijia Lin, Sicheng Zhao, Jungong Han, Large-Scale Image Retrieval with Sparse Embedded Hashing, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.01.055

Large-Scale Image Retrieval with Sparse Embedded Hashing

Guiguang Ding[a,*], Jile Zhou[a], Yuchen Guo[a], Zijia Lin[a], Sicheng Zhao[a], Jungong Han[b]

[a]*Intelligent Multimedia Group, School of Software, Tsinghua University, Beijing, China*
[b]*Department of Computer Science and Digital Technologies, Northumbria University, Newcastle, UK*

Abstract

In this paper, we present a novel sparsity-based hashing framework termed Sparse Embedded Hashing (SEH), exploring the technique of sparse coding. Unlike most of the existing systems that focus on finding either a better sparse representation in hash space or an optimal solution to preserve the pairwise similarity of the original data, we intend to solve these two problems in one goal. More specifically, SEH firstly generates sparse representations in a data-driven way, and then learns a projection matrix, taking sparse representing, affinity preserving and linear embedding into account. In order to make the learned compact features locality sensitive, SEH employs the matrix factorization technique to approximate the Euclidean structures of the original data. The usage of the matrix factorization enables the decomposed matrix to be constructed from either visual or textual features depending on which kind of Euclidean structure is preserved. Due to this flexibility, our SEH framework could handle both single-modal retrieval and cross-modal retrieval simultaneously. Experimental evidence shows this method achieves much better performance in both single- and cross-modal retrieval tasks as compared to state-of-the-art approaches.

Keywords: Hashing; Sparse Coding; Matrix Factorization

*Corresponding author
Email address: dinggg@tsinghua.edu.cn (Guiguang Ding)

1. Introduction

Nearest Neighbor (NN) retrieval, a method of finding the semantically nearest item to a query item from a search database, is facing efficiency problem due to the explosive growth of data on the Internet. *Approximate Nearest Neighbor* (ANN) search is a more efficient alternative technique that well balances the accuracy and the computational complexity.

As the most notable ANN method, hashing technique aims to convert the high-dimensional data item to a short code consisting of a sequence of binary bits while preserving the similarity between the original data points [1, 2, 3, 4, 5]. Hashing can deal with ANN search efficiently because bit XOR and bit-count operations are applied when calculating Hamming distance between binary codes [6]. This technique has shown to be useful for many practical problems, thus gaining considerable attention in the field of large-scale image retrieval in the past decade.

Generally, hashing methods can be divided into two categories: single-modal hashing (SMH) and cross-modal hashing (CMH). The majority of the existing works fall into the category of SMH which is designed for uni-modal data. As the most well-known SMH approach, Locality-Sensitive Hashing (LSH) [7] simply employs random linear projections to map high-dimensional features into a binary sequance such that the close features in Euclidean space still remain to be close after the transformation. Although this technique has been exploited in various applications, it is likely to generate ineffective codes due to its data-independent property [6]. Hence, some machine learning techniques that learn the data characteristics have been employed to design more effective hash functions, such as Kernel Learning, Boosting algorithm, Restricted Boltzmann Machines, Manifold Learning, Supervised Learning, Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), which respectively correspond to Kernelized Hashing [8, 9], Parameter Sensitive Hashing [10], Semantic Hashing [11], Spectral Hashing [12, 13], Supervised Hashing [14], LDA Hashing [15], PCA Hashing [16] and K-means Hashing [17].

At the early stage, hashing methods are only applied to unimodal data. As the fast growth of multimedia content on the Web, like Wikipedia, Flickr and Twitter, the cross-modal hashing (CMH), returning semantically relevant results of the other modalities for a given query from one modality, is in great demand. For instance, Wikipedia is a popular dataset consisting of images and texts. Usually, the system allows users to provide a query text, and it returns relevant texts as well as pictures. However, users, very often, prefer to provide a query image without texts but expect the system to return the relevant articles. Such practical requests thus boost the research in the field of cross-modal content search [18, 19, 20, 21, 22, 23, 24, 25].

1.1. Motivation

The key of hashing based data retrieval is to capture salient structures and meanwhile to preserve the similarity of the original data points. Recently, sparse coding has been adopted to address large-scale data retrieval problem for both single modality and cross modalities [26, 27, 28, 29, 30, 31, 32] due to the following reasons. First, the natural image can be well described based on a small number of structural primitives [33, 34, 35, 36]. Second, the sparsity constraint allows the learned representation to capture salient structures of the image [37, 38, 39]. Finally, sparse coding can be applied to learn over-complete bases, which provides sufficient descriptive power for representing low-level features [40, 30].

Despite the increasing research interest from the academia, the results obtained by the existing sparse coding hashing attempts are still far from satisfactory. The major reason is the lack of the solution which could simultaneously address the following three problems:

- how to embed sparse representations into a compact space to generate hash codes?
- how to preserve the similarity structures of the original data?

- how to cope with both single-modal retrieval and cross-modal retrieval in one system?

Most existing hashing methods only partially addressed the first two problems, and they are designed specifically for either single-modality retrieval or multiple-modality retrieval. For instance, Robust Sparse Hashing [26], Compact Sparse Codes [30] and Sparse Multimodal Hashing [29] advocate the use of compact codes by encoding sparse codes into a set of integers. Although the generated compact codes well preserve the original similarity structure, they are less efficient than binary codes in terms of the storage space and the searching cost. Sparse Hashing [27] indeed generates binary codes by setting each non-zero value of sparse codes to be 1. However, such a simple binarization rule is unable to generate balanced codes. Compressed Hashing [28] embeds sparse codes using the random projection technique, leading to ineffective codes because of its data-independence nature. In addition, these sparsity-based hashing methods adopt two-step solutions that separate the sparse codes leaning and embedding, which can only achieve suboptimal results.

1.2. Contributions

In this paper, we introduce a novel sparsity-based hashing framework, namely Sparse Embedded Hashing (SEH), intending to address the above three problems simultaneously via optimizing an objective funcition that takes all of aboves into account. Our work differs from existing systems in two aspects. First, instead of using a two-step approach, we consider sparse representing, affinity preserving and linear embedding in one objective function when learning the projection matrix. Second, in order to make the learned compact features locality sensitive, SEH employs the matrix factorization technique to approximate the Euclidean structures of the original data. We theoretically prove that the matrix factorization technique relaxes the orthogonality constraints and is better suited to preserve the similarity of data points than commonly used PCA technique. In addition, the decomposed matrix can be constructed from either visual or textual features depending on which kind of similarity structure is preserved. Due to this flexibility, our SEH could handle both single-modal retrieval and cross-modal retrieval in one system.

The rest of this paper is organized as follows. We formulate several related cross-modal hashing methods and Canonical Correlation Analysis (CCA) within the same framework in Section 2. Section 3 presents our proposed method. Section 4 provides extensive experimental validation on three datasets. The conclusions are given in Section 5.

2. Related Work

As our major contribution is a new methodology that incorporates the sparse coding into image hashing, we focus on explaining sparse coding related image hashing techniques. Here, we start by presenting the original sparse representation idea that can be used in a variety of applications such as image classification [41], face recognition [42], image denoising [43] and image restoration [44]. Afterwards, we elaborate the existing sparse hashing algorithms.

2.1. Sparse Coding

Let $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ is the data vector, $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_D] \in \mathbb{R}^{d\times D}$ is the codebook, where each $\mathbf{b}_i$ is a basis vector. $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_n] \in \mathbb{R}^{D\times n}$ denotes the coefficient matrix, in which each column is a sparse representation. Given a data point $\mathbf{x}_i$, it can be approximated by linearly combining a small number of (sparse) basis vectors in the codebook, i.e. $\mathbf{x}_i \approx \mathbf{B}\mathbf{s}_i$. Typically, ℓ_2 norms, i.e. sum of square value of each entry in matrix or vector, is used for measuring the loss function of the reconstruction error, which is:

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{B}\mathbf{s}_i\|_2^2.$$

Then, the objective function of sparse coding can be formulated as follows:

$$\min_{\mathbf{B},\mathbf{S}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{B}\mathbf{s}_i\|_2^2 + \lambda \sum_{i=1}^{n} f(\mathbf{s}_i),$$

where f is a function to measure the sparsity of $\mathbf{s}_i$, and $\lambda > 0$ is the tunable regularization parameter controlling the sparsity. For example, we can use one of the following penalty functions [45]:

$$f(\mathbf{s}_i) = \sum_{j=1}^{D} \begin{cases} \|s_{ij}\|_{\ell_1} & (\ell_1 \text{ penalty function}) \\ (s_{ij}^2 + \epsilon)^{\frac{1}{2}} & (\text{epsilon } \ell_1 \text{ penalty function}) \\ log(1 + s_{ij}^2) & (\text{log penalty function}), \end{cases}$$

where $\|\cdot\|_{\ell_1}$ denotes ℓ_1-norm, i.e. sum of the absolute value of each entry in a matrix or a vector. In this paper, we concentrate on the case of ℓ_1 penalty function, because it is known to produce sparse coefficients and can be robust to irrelevant features [46]. Then, the objective function becomes:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{S}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{B}\mathbf{s}_i\|_2^2 + \lambda \sum_{i=1}^{n} \|\mathbf{s}_i\|_{\ell_1} \\ &s.t.\ \|\mathbf{b}_j\| \leq 1, \forall j \in \mathcal{I}_D, \end{aligned} \tag{1}$$

where $\mathcal{I}_D = \{1, 2, ..., D\}$ is the index set. The constraint on $\mathbf{b}_j$ is typically applied to avoid trivial solutions.

2.2. Locality-sensitive Sparse Coding

Usually, the codebook $\mathbf{B}$ is over completed, i.e. $D > d$. In this case, the ℓ_1 regularization is to ensure that the Eq. (1) has a unique solution. However, due to the over-completeness of the codebook, the sparse coding process may find different bases for similar data vectors, thus losing correlations between codes [37]. In [47], the authors pointed out that locality is more important than sparsity under certain assumptions [48]. To this end, generating locality sensitive sparse codes has been investigated in several works [37, 30, 49, 50, 26], each being elaborated below.

Graph Laplacian Sparse Coding [49, 50] intends to generate similar sparse codes for similar local features $\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^n$. Such an idea can be implemented by adding the following Laplacian regularization into Eq. (1),

$$\frac{1}{2} \sum_{i,j=1}^{n} \mathbf{W}_{ij} \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 = \sum_{i,j=1}^{n} \mathbf{L}_{ij} \mathbf{s}_i^T \mathbf{s}_j = \mathrm{tr}(\mathbf{S}\mathbf{L}\mathbf{S}^T). \tag{2}$$

Here, $\mathbf{W} \in \mathbb{R}^{n\times n}$ is the similarity matrix, in which $\mathbf{W}_{ij}$ refers to the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathrm{tr}(\cdot)$ denotes the trace function. $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix, and $\mathbf{D}$ is a diagonal degree matrix subject to $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{W}_{ij}$. Therefore, we can get the following objective function of graph Laplacian sparse coding,

$$\min_{\mathbf{B},\mathbf{S}} \|\mathbf{X} - \mathbf{BS}\|_F^2 + \lambda \|\mathbf{S}\|_{\ell_1} + \beta \mathrm{tr}(\mathbf{SLS}^T)$$
$$s.t.\ \|\mathbf{b}_j\| \leq 1, \forall j \in \mathcal{I}_D,$$

where $\beta > 0$ is the regularization parameter, and $\|\cdot\|_F$ is the Frobenius norm.

Robust Sparse Hashing (RSH) [26], in order to be robust against the random perturbations, seeks a dictionary such that all points $\mathbf{x} \in \mathcal{U}_P(\widehat{\mathbf{x}}) = \{\mathbf{x}; \|\mathbf{P}(\mathbf{x} - \widehat{\mathbf{x}})\| < 1\}$ tend to have the same hash codes, where $\mathbf{P}$ is positive definite matrix. The objective function of RSH can be described by:

$$\min_{\mathbf{B},\mathbf{S}} \|\mathbf{X} - \mathbf{BS}\|_F^2 + \lambda \|\mathbf{S}\|_{\ell_1}$$
$$s.t.\ \|\mathbf{b}_j\| \leq 1, \mathbf{x}_i \in \mathcal{U}_P(\widehat{\mathbf{x}}_i), \forall j \in \mathcal{I}_D, \forall i \in \mathcal{I}_n.$$

Locality-constrained Linear Coding (LLC) [37] utilizes the locality constraints to project each descriptor into its local-coordinate system, and the projected coordinates are regarded as sparse codes. Basically, the LLC code uses the below criteria:

$$\min_{\mathbf{B},\mathbf{S}} \|\mathbf{X} - \mathbf{BS}\|_F^2 + \lambda \sum_{i=1}^{n} \|\mathbf{d}_i \odot \mathbf{c}_i\|_2^2$$
$$s.t.\ \mathbf{1}^T \mathbf{c}_i = 1, \forall i \in \mathcal{I}_n,$$

where $\odot$ denotes the element-wise multiplication, and $\mathbf{d}_i$ is defined as:

$$\mathbf{d}_i = exp(\frac{d_E(\mathbf{x}_i, \mathbf{B})}{\sigma}).$$

Here, $d_E(\mathbf{x}_i, \mathbf{B}) = [d_E(\mathbf{x}_i, \mathbf{b}_1), ..., d_E(\mathbf{x}_i, \mathbf{b}_D)]^T$, and $d_E(\mathbf{x}_i, \mathbf{b}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{b}_j$. σ is used for adjusting the weight decay speed for the locality adapter.

Compact Sparse Codes (CSC) [30] theoretically proves that the sensitivity of sparse codes is related to the coherence of the dictionary. To this end, CSC integrates the incoherence constraint of codebook into the sparse coding objective function as follows:

$$\min_{\mathbf{B},\mathbf{S}} \|\mathbf{X}-\mathbf{BS}\|_F^2 + \lambda\|\mathbf{S}\|_{\ell_1}$$
$$s.t.\ \|\mathbf{B}_{\sim k}^T \mathbf{b}_k\|_\infty \leqslant \gamma; \forall k \in \mathcal{I}_n,$$

where $\mathbf{B}_{\sim k}^T$ means the codebook $\mathbf{B}$ with the k-th column removed, and γ is a constant; $\mu_{min} \leqslant \gamma \leqslant 1$ controls the allowed dictionary coherence, and μ_{min} is the minimum coherence of dictionary $\mathbf{B}$; $\|\cdot\|_\infty$ corresponds to the maximum absolute value of entries in an input vector.

2.3. Affinity-preserving Embedding

Similar to the request for locality-sensitive sparse codes, generating compact features via sparse codes that preserve the affinity of the original data is also important, and it has been well recognized by the researchers in this field. Some representatives proposed recently include [26, 28, 27, 51, 29, 52, 30].

Sparse Hashing [27] and Sparse Multimodal Hashing (SMH) [29] generate compact binary codes by simply setting each non-zero entry of the sparse codes to be 1. However, there are two issues attached to this binarization rule. Firstly, it fails to build compact binary codes, because over-complete basis (i.e. large dictionary size D) is always applied in sparse coding for sufficient descriptive power [40, 30]. Secondly, the ℓ_1-norm penalty function guarantees the coefficients $\mathbf{s}$ in Eq. (1) to be sparse. Hence, the number of zero entries is far greater than the number of non-zero entries in a sparse representation (empirically, more than 90% entries are zero in $\mathbf{s}$), which leads to unbalanced binary codes.

RSH [26] and CSC [30] encode sparse codes into a set of integers, which are composed of non-zero indexes $J(\mathbf{x}) = \{j; \mathbf{s}_j(\mathbf{x}) \neq 0, j \in \mathcal{I}_D\}$, where $\mathbf{s}_j(\mathbf{x})$ is the j-th atom in sparse code of $\mathbf{x}$. The similarity between index set J_i and J_j is measured by Jacard distance, which is $|J_i \cap J_j|/|J_i \cup J_j|$. In reality, Jacard distance can be approximated by using Min-Hash [53]. Apparently, the index set does not have the advantages of efficient storage and bitwise operations anymore as compared against binary codes.

Sparse-Coded Features (SCF) [51] and CH [28] embed sparse representation $\mathbf{s}(\mathbf{x})$ into a low-dimensional space by a reduction matrix $\mathbf{P} \in \mathbb{R}^{k \times D}$ which satisfies $k < D$:

$$\mathbf{z}(\mathbf{x}) = \mathbf{P}\mathbf{s}(\mathbf{x}). \tag{3}$$

Here, SCF constructs $\mathbf{P}$ by selecting the largest k eigenvalues of covariance matrix $\mathbf{S}\mathbf{S}^T$ (i.e. PCA) whereas CH independently samples each entry $\mathbf{P}_{ij}$ from a Gaussian distribution $\mathcal{N}(0, 1/k)$. Similar to PCA, SCF embeds sparse codes into compact features space but tries to preserve global Euclidean structures of the sparse space. The details will be discussed in section 3.2.3. Also, according to Restricted Isometry Property (RIP), for any integer $t > 0$, if t/D is small enough and $k = ct\log(D/t)$, where c is a constant, there exists a positive constant $\delta_t < 1$ such that with an overwhelming probability, the following inequality holds for any $\mathbf{s} \in \mathbb{R}^D$ with at most t non-zero entries [28]

$$(1 - \delta_t)\|\mathbf{s}\|_2^2 \leq \frac{D}{k}\|\mathbf{z}\|_2^2 \leq (1 + \delta_t)\|\mathbf{s}\|_2^2. \tag{4}$$

Inequality in Eq. (4) shows that RIP assures to preserve its Euclidean structures when mapping the sparse code $\mathbf{s}$. Actually, these methods firstly learn local sparse codes, and then embed them into a compact space by an affinity-preserving transformation. Such a two-step solution gives rise to suboptimal results. In contrast to these methods, we simultaneously consider sparse coding and affinity-preserving embedding in order to seek the best trade-off.

At the last stage of hash function learning, several quantization algorithms (such as Graph Hashing [54], ITQ [55, 56], Double Bit Hashing [57] and K-means Hashing [17]) can be selected to quantify the embedded compact features into binary space. However, this is not the focus of our work. Therefore, we simply regard quantization strategy as a sign function, in which $\text{sign}(v) = 1$ if $v \geq 0$, and -1 otherwise.

3. Sparse Embedded Hashing

A flowchart of our Sparse Embedded Hashing framework is given in Fig. 1. Given a new query $\mathbf{x}^*$, SEH obtains its binary hash codes $h(\mathbf{x}^*)$ by pre-trained

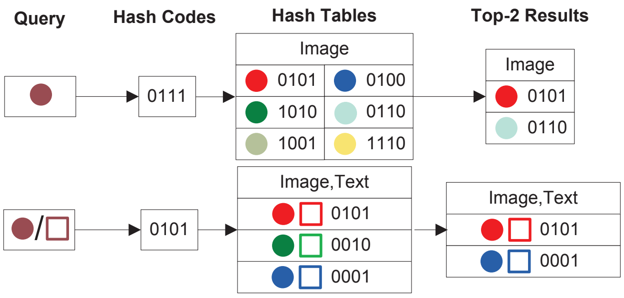


Figure 1: Flowchart of SEH, where circle and square denote image and text respectively, illustrated with toy data. Top) SEH deals with single-modal retrieval (SMR). Bottom) SEH learns unified hashcodes for each modality of data in the task of cross-modal retrieval (CMR).

hash function h, then scans over the hash table linearly, and eventually returns similar results for the given mapped query (Fig. 1 Top). If the semantic text feature $\mathbf{y}_i \in \mathbb{R}^d$ is available, e.g. a sample consisting of an image and its surrounding text ($o_i = (\mathbf{x}_i, \mathbf{y}_i), i \in \mathcal{I}_n$), SEH could learn an integrated binary code for both modalities. As illustrated in Fig. 1 Bottom, SEH maps a query (image or text) to a common Hamming space, then returns semantically relevant results of the other modalities, facilitating cross-modal retrieval. SEH is suitable for an online large-scale data search task, since only bit XOR operations are performed when calculating Hamming similarities between binary codes.

3.1. Problem Formulation

Let us now introduce a set of notations. Assume that $\mathcal{O} = \{o_i\}_{i=1}^n$ is a set of samples with $\mathbf{x}_i \in \mathbb{R}^m$ being the i-th image descriptor of $\mathcal{O}$. Given the hash code length k, the purpose of SEH is to learn hash functions $\{h_j\}_{j=1}^k$, which map original data in $\mathbb{R}^m$ to a Hamming space[1] $\{0, 1\}^k$ with $h(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_k(\mathbf{x})]^T$. Actually, the function h can be decomposed as follows:

$$h(\mathbf{x}) = q[g(\mathbf{x})],$$

[1]It is equivalent to denote $\{-1, 1\}^k$ as Hamming space via a linear transformation.

where $g : \mathbb{R}^m \to \mathbb{R}^k$ is the real-valued embedding function, and $q : \mathbb{R}^k \to \{0,1\}^k$ is the quantization function. As mentioned above, we simply set $q(\mathbf{x}) = \text{sign}(\mathbf{x})$.

3.2. Objective Function

The core of hashing based image retrieval is the goal of preserving similarity of original data and capturing salient structures of image. Hence, SEH generates sparse codes $\mathbf{s}_i$ for each image descriptor $\mathbf{x}_i$ via over-complete bases so as to sufficiently capture structural primitives of image. However, the learned sparse codes $\{\mathbf{s}_i\}_{i=1}^n$ are neither compact nor locality-sensitive. Our SEH solves this problem using two different approaches. On the one hand, in order to obtain compact feature $\mathbf{z}_i$, SEH considers the embedding projection of the form as suggested in Eq. (3), i.e. $\mathbf{z}_i = g(\mathbf{x}_i) = \mathbf{P}\mathbf{x}_i$. On the other hand, in order to make the learned compact features (i.e. $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_n]$) locality-sensitive, SEH uses matrix factorization to approximate the Euclidean structures of the original data (i.e. $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$). Different from existing two-step sparsity-based hashing methods that only achieve suboptimal results, SEH integrates sparse coding, compact embedding and similarity preserving together and solves these three problems in one objective function. An iterative strategy is designed to explore the optimal solution for SEH. Finally, the hash code is obtained by quantization function $q(\mathbf{z}_i)$. Before presenting our overall objective function, we first look into these three subproblems separately.

3.2.1. Sparse Coding

Data-dependent sparse coding, describing each sample based on only several active vectors of trained dictionary, has been popularly utilized as an effective image representation in many applications. As mentioned above, we concentrate on the case of ℓ_1 regularization to control the sparsity as shown in Eq. (1), and rewrite it to the matrix form as follows:

$$\mathcal{L}_{\text{sc}}(\mathbf{B}, \mathbf{S}) = \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 + \lambda \|\mathbf{S}\|_{\ell_1}. \tag{5}$$

We let $\mathbf{B}$ be over-complete (i.e. $D > d$), because it provides sufficient descriptive power for low-level features of image [40, 30]. Actually, the optimal solution $\mathbf{S}^*$ in Eq. (1) is sparse but perturbation sensitive [49]. Next, we will present how to embed $\mathbf{S}^*$ into compact space while preserving the similarity structures of the original data.

200 *3.2.2. Compact Embedding*

We consider the embedding projection of the form as suggested in Eq. (3), and reformulate it using matrix form:

$$\mathbf{Z} = \mathbf{P}\mathbf{S}. \tag{6}$$

It may end up with infinitely many solutions $\mathbf{P}$ satisfying the Eq. (6) (given $\mathbf{Z}$ and $\mathbf{S}$), because $\mathbf{S}$ is not reversible. Fortunately, $\mathbf{P}$ can be approximated by minimizing the following quadratic equation,

$$\mathcal{L}_{\text{em}}(\mathbf{P}) = \|\mathbf{Z} - \mathbf{P}\mathbf{S}\|_F^2. \tag{7}$$

The smaller $\mathcal{L}_{\text{em}}(\mathbf{P})$ usually means the better approximation solution, and the optimization problem $\min_{\mathbf{P}}\{\mathcal{L}_{\text{em}}(\mathbf{P})\}$ can be easily solved through matrix derivative operation.

3.2.3. Similarity Preserving

As mentioned above, preserving the similarity structures of the original data is a key issue in the process of hash function learning. Normally, PCA, the most notable low-dimensional embedding technique, is employed to preserve the global structures of original data, which can be briefly recapped below. Denote $\mathbf{w}_t$ and λ_t as the t-th eigenvector and eigenvalue of $\mathbf{X}\mathbf{X}^T$ respectively, According to the definition of eigenvector and eigenvalue, we have,

$$\mathbf{X}\mathbf{X}^T\mathbf{w}_t = \lambda_t\mathbf{w}_t. \tag{8}$$

Suppose $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_d]$, we can get the following formula,

$$\begin{aligned} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{W}\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j) \\ &= \|\mathbf{x}_i - \mathbf{x}_j\|_2^2. \end{aligned} \tag{9}$$

Eq. (9) holds because $\mathbf{W}$ is orthogonal[2], i.e. $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^\mathbf{T} = \mathbf{I}$. The largest k eigenvectors are selected as principal components in PCA. With $\mathbf{W}_1 = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k]$, the PCA embedding is performed as

$$\mathbf{Z} = \mathbf{W}_1^T\mathbf{X} \quad \text{or} \quad \mathbf{X} = \mathbf{W}_1\mathbf{Z}. \tag{10}$$

Now we'd like to investigate the global structure preserving of PCA. According to Eq. (9) and Eq. (10) above, we have

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 &= \|\mathbf{W}_1^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \|\mathbf{W}_2^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \\ &= \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 + \|\mathbf{W}_2^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2, \end{aligned} \tag{11}$$

where $\mathbf{W}_2 = [\mathbf{w}_{k+1}, ..., \mathbf{w}_d]$. Obviously, by the triangle inequality and non-negativity properties of norm, we can get the following inequalities,

$$0 \leq \|\mathbf{W}_2^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \leq \|\mathbf{W}_2^T\mathbf{x}_i\|_2^2 + \|\mathbf{W}_2^T\mathbf{x}_j\|_2^2. \tag{12}$$

Denote $\epsilon_i = \|\mathbf{W}_2^T\mathbf{x}_i\|_2^2$, and substitute Eq. (12) into Eq. (11), then we can get the bounds of $\|\mathbf{z}_i - \mathbf{z}_j\|_2$ as:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 - (\epsilon_i + \epsilon_j) \leq \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2^2. \tag{13}$$

It's necessary to analyze the expectation of ϵ_i in depth. Assume that each descriptor $\mathbf{x}_i$ is sampled uniformly, hence we have

$$\begin{aligned} \mathbb{E}(\epsilon) \approx \sum_i \epsilon_i/n &= \sum_i \|\mathbf{W}_2^T\mathbf{x}_i\|_2^2/n \\ &= \sum_{t=k+1}^{d} \sum_i (\mathbf{w}_t^T\mathbf{x}_i)^2/n \\ &= \sum_{t=k+1}^{d} \mathbf{w}_t^T\mathbf{X}\mathbf{X}^T\mathbf{w}_t/n. \end{aligned} \tag{14}$$

Substituting Eq. (8) into Eq. (14) will lead to:

$$\mathbb{E}(\epsilon) \approx \sum_{t=k+1}^{d} \lambda_t\mathbf{w}_t^T\mathbf{w}_t/n \propto \sum_{t=k+1}^{d} \lambda_t. \tag{15}$$

[2] Because the eigenvector of symmetrical matrix $\mathbf{X}\mathbf{X}^T$ is orthogonal, i.e. $\mathbf{w}_i^T\mathbf{w}_j = \delta_{ij}, i, j \in \mathcal{I}_d$, where δ_{ij} is Kronecker delta, and it is 1 if the variables are equal, and 0 otherwise.

Eq. (15) implies that the approximated expectation of ϵ is proportional to the summation of the last $d-k$ eigenvalue of $\mathbf{X}\mathbf{X}^T$. Actually, Eq. (15) also reveals that selecting the largest k eigenvectors as the principal components (PCA technique) essentially minimizes the approximate expectation of ϵ.

However, Wang et al. prove that the orthogonality of embedding matrix (i.e. $\mathbf{W}_1^T\mathbf{W}_1 = \mathbf{I}$) actually degrades the performance of a CBIR system, because the low-variance directions will be picked up when a long code is required [58]. Hence, in our algorithm, we relax the orthogonality constraints in PCA embedding Eq. (10), allowing successive projections to capture more of the data variance. Analogous to Eq. (7) mentioned in the previous section, Eq. (10) can be approximated by minimizing the following quadratic equation without an orthogonality regularization, so

$$\mathcal{L}_{\text{ap}}^{(X)}(\mathbf{W}, \mathbf{Z}) = \|\mathbf{X} - \mathbf{W}\mathbf{Z}\|_F^2, \tag{16}$$

where $\mathbf{W} \in \mathbb{R}^{d\times k}$ is the embedding matrix. Again, it is required to investigate whether the global structure can be preserved by solving Eq. (16). Here, $\mathbf{W}$ tends to be a full rank matrix because usually $k \ll d$, and if the factorization is perfect (i.e. $\mathbf{X} = \mathbf{W}\mathbf{Z}$), we could obtain two important inequalities as follows,

$$\|\mathbf{W}\|^{-1}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \le \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \le \|\widehat{\mathbf{W}}\|\|\mathbf{x}_i - \mathbf{x}_j\|_2^2, \tag{17}$$

where $\widehat{\mathbf{W}}$ is the left inverse of $\mathbf{W}$, i.e. $\widehat{\mathbf{W}}\mathbf{W} = \mathbf{I}$. Compared to inequalities in Eq. (13), the inequalities described in Eq. (17) control the bounds of $\|\mathbf{z}_i - \mathbf{z}_j\|_2^2$ through $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ multiplied by a constant[3]. Minimizing Eq. (16) would reduce the reconstruction error of matrix factorization (usually, not equal to 0) which affects the bounds significantly. Empirically, we investigate the distribution of reconstruction error of matrix factorization based on two large datasets. The results in section 4.2 also reveal that the error is always small in real applications. Furthermore, we compare SEH with several state-of-the-arts hashing methods on a public dataset (SIFT1M), which is usually used to evaluate the ANN

[3] Actually, Inequalities (17) is known as bi-Lipschitz continuity in *mathematical analysis*.

search performances [59]. The results consistently reflect the superior ability of similarity-preserving of SEH.

In addition, if the semantic text $\mathbf{y}_i$ is available, we can also use $\mathbf{Z}$ to approximate the structures of $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n]$, which may be more precise. Similarly, we have

$$\mathcal{L}_{ap}^{(Y)}(\mathbf{W}, \mathbf{Z}) = \|\mathbf{Y} - \mathbf{W}\mathbf{Z}\|_F^2. \tag{18}$$

In fact, each column vector $\mathbf{z}_i^*$ of the optimal solution in Eq. (18) is the k-dimensional representation in latent semantic space [60, 61]. There is an intuitive interpretation about combining Eq. (6) and Eq. (18) together, which is actually a latent concept described by several image salient structures [24].

To sum up, either Eq. (16) or Eq. (18) is able to control Euclidean structure approximating in the proposed approach, i.e. similarity preserving. To our best knowledge, this is the first attempt to explore the matrix factorization for similarity preserving.

3.2.4. Overall Objective Function

The overall objective function, combining the sparse representing, affinity preserving and linear embedding together, is defined by:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{P},\mathbf{W},\mathbf{Z},\mathbf{S}} \mathcal{L}(\mathbf{B}, \mathbf{P}, \mathbf{W}, \mathbf{Z}, \mathbf{S}) = \mathcal{L}_{\text{sc}} + \mu \mathcal{L}_{\text{em}} + \gamma \mathcal{L}_{\text{ap}}^{(\cdot)} \\ &s.t. \|\mathbf{b}_i\|_2^2 \leq 1, \|\mathbf{p}_j\|_2^2 \leq 1, \|\mathbf{w}_t\|_2^2 \leq 1, i, j \in \mathcal{I}_D, t \in \mathcal{I}_k, \end{aligned} \tag{19}$$

where $\mu, \gamma > 0$ are the fixed weight parameters and we will experimentally investigate how system performance will behave when varying those parameters in section 4.5. $\mathcal{L}_{\text{ap}}^{(\cdot)}$ denotes either $\mathcal{L}_{\text{ap}}^{(X)}$ or $\mathcal{L}_{\text{ap}}^{(Y)}$, and $\|\cdot\|_2^2 \leq 1$ is applied to avoid trivial solution.

3.3. Optimization Algorithm

Optimizing Eq. (19) is basically a non-convex problem, because there are five matrix variables $\mathbf{B}, \mathbf{Z}, \mathbf{P}, \mathbf{W}, \mathbf{S}$. Fortunately, it becomes convex with respect to any one of the five variables while fixing the other four. Therefore, the optimization problem can be solved by the following listed steps iteratively until its convergence. Actually, no matter what type of $\mathcal{L}_{\text{ap}}^{(\cdot)}$ is, the solution for optimizing Eq. (19) is essentially identical, therefore we only take $\mathcal{L}_{\text{ap}}^{(Y)}$ as an example.

Step1: Learning sparse representations $\mathbf{S}$ by fixing the other variables, then Eq. (19) w.r.t. $\mathbf{S}$ is written as follows

$$\begin{aligned} \min_{\mathbf{S}} \ \mathcal{L}(\mathbf{S}) &= \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_{\ell_1} + \gamma||\mathbf{Z} - \mathbf{P}\mathbf{S}\|_F^2 \\ &= \| \begin{bmatrix} \mathbf{X} \\ \sqrt{\gamma}\mathbf{Z} \end{bmatrix} - \begin{bmatrix} \mathbf{B} \\ \sqrt{\gamma}\mathbf{P} \end{bmatrix} \mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_{\ell_1}. \end{aligned} \tag{20}$$

245 We solve the ℓ_1-norm regularized least square problem by SLEP (Sparse Learning with Efficient Projections) package[4].

Step2: Again, learning compact embedded features $\mathbf{Z}$ by fixing the others variables, then Eq. (19) is rewritten as:

$$\min_{\mathbf{Z}} \ \mathcal{L}(\mathbf{Z}) = \mu\|\mathbf{Y} - \mathbf{W}\mathbf{Z}\|_F^2 + \gamma\|\mathbf{Z} - \mathbf{P}\mathbf{S}\|_F^2. \tag{21}$$

By taking the derivative of Eq. (21) with respect to $\mathbf{Z}$,

$$\frac{\partial\mathcal{L}(\mathbf{Z})}{\partial\mathbf{Z}} = 2\mu\mathbf{W}^T(\mathbf{Y} - \mathbf{W}\mathbf{Z}) + 2\gamma(\mathbf{Z} - \mathbf{P}\mathbf{S}), \tag{22}$$

and setting Eq. (22) to 0, we can obtain the close-form solution, which is

$$\mathbf{Z} = (\mathbf{W}^T\mathbf{W} + \frac{\gamma}{\mu}\mathbf{I})^{-1}(\frac{\gamma}{\mu}\mathbf{P}\mathbf{S} + \mathbf{W}^T\mathbf{Y}). \tag{23}$$

Step3: Learning $\mathbf{B}, \mathbf{P}, \mathbf{W}$ respectively using the Lagrange dual [45]. In fact, the learning problem w.r.t. $\mathbf{B}, \mathbf{P}, \mathbf{W}$ is essentially identical, hence we only show how to optimize $\mathbf{B}$ as the example. Fixing other variables, the Eq. (19) becomes the least squares problem with quadratic constraints:

$$\begin{aligned} &\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 \\ &s.t. \|\mathbf{b}_i\|_2^2 \leq 1, i \in \mathcal{I}_D. \end{aligned} \tag{24}$$

[4]http://parnec.nuaa.edu.cn/jliu/largeScaleSparseLearning.htm

Algorithm 1 Sparse Embedded Hashing

Input:

Training matrix $\mathbf{X}, \mathbf{Y}$, parameters λ, μ, γ, bit number k

Output:

Hash codes $\mathbf{H}$, matrix variables $\mathbf{B}, \mathbf{W}, \mathbf{Z}$.

1: Initialize $\mathbf{Z}, \mathbf{W}, \mathbf{P}$ and $\mathbf{B}$ by random matrices respectively, and normalizing each column of $\mathbf{X}$ by ℓ_2 norm.

2: **repeat**

3: Fix $\mathbf{Z}, \mathbf{P}, \mathbf{B}$ and $\mathbf{W}$, update $\mathbf{S}$ as illustrated in Step1;

4: Fix $\mathbf{W}, \mathbf{P}, \mathbf{B}$ and $\mathbf{S}$, update $\mathbf{Z}$ by Equation (23);

5: Fix $\mathbf{W}, \mathbf{P}, \mathbf{W}$ and $\mathbf{S}$, update $\mathbf{B}$ as illustrated in Step3;

6: Fix $\mathbf{Z}, \mathbf{B}, \mathbf{W}$ and $\mathbf{S}$, update $\mathbf{P}$ by optimizing:

$$\begin{aligned} &\min_{\mathbf{P}} \ \|\mathbf{Z} - \mathbf{P}\mathbf{S}\|_F^2 \\ &s.t. \ \|\mathbf{p}_i\|_2^2 \leq 1, i \in \mathcal{I}_D \end{aligned}$$

7: Fix $\mathbf{P}, \mathbf{B}, \mathbf{Z}$ and $\mathbf{S}$, update $\mathbf{W}$ by optimizing:

$$\begin{aligned} &\min_{\mathbf{W}} \ \|\mathbf{Y} - \mathbf{W}\mathbf{Z}\|_F^2 \\ &s.t. \ \|\mathbf{w}_i\|_2^2 \leq 1, i \in \mathcal{I}_k \end{aligned}$$

8: **until** convergency.

9: $\mathbf{H} = \text{sign}(\mathbf{Z})$.

Consider the Lagrangian:

$$\mathcal{L}(\mathbf{B}, \vec{\theta}) = \|\mathbf{X} - \mathbf{B}\mathbf{S}\|_F^2 + \sum_{i=1}^{n} \theta_i (\|\mathbf{b}_i\|_2^2 - 1), \tag{25}$$

where $\theta_i > 0$ is the Lagrange multipliers. Setting the derivative of Eq. (25) w.r.t. $\mathbf{B}$ to be zero, the close form solution for Eq. (24) is

$$\mathbf{B} = \mathbf{X}\mathbf{S}^T (\mathbf{S}\mathbf{S}^T + \mathbf{\Theta})^{-1}, \tag{26}$$

where $\mathbf{\Theta}$ is a diagonal matrix with diagonal entry being $\mathbf{\Theta}_{ii} = \theta_i$, which can be obtained by optimizing the following Lagrange dual problem

$$\begin{aligned} &\min_{\mathbf{\Theta}} \ \mathrm{tr}(\mathbf{X}\mathbf{S}^T(\mathbf{S}\mathbf{S}^T + \mathbf{\Theta})^{-1}\mathbf{S}\mathbf{X}^T) + \mathrm{tr}(\mathbf{\Theta}). \\ &s.t. \ \mathbf{\Theta}_{ii} \geq 0, i \in \mathcal{I}_D \end{aligned} \tag{27}$$

Eq. (27) can be solved by using Newtons method or conjugate gradient. The complete algorithm is summarized in Alg. 1.

3.4. Computational Complexity Analysis

Typically, solving (20) and (21) requires $O(nM^2)$[5] and $O(d^3)$ respectively. The Lagrange dual (27), which is independent to n, can be solved by using Newtons method or conjugate gradient, which show better efficiency than steepest gradient descent [45]. In a word, the total time complexity of training SEH is linear to n, which is really scalable for large-scale datasets compared with most existing data-dependent hashing.

4. Experiments

In this section, we evaluate the ANN search performances in similarity-preserving, single- and cross-modal retrieval tasks, respectively.

4.1. Experiment Settings

4.1.1. Evaluation Metrics

First of all, we introduce two basic metrics that we used to measure the system performance, which are:

$$\begin{aligned} \text{Precision} &= \frac{\#\text{relevant instance retrieved}}{\#\text{retrieved instance}} \\ \text{Recall} &= \frac{\#\text{relevant instance retrieved}}{\#\text{all relevant instance}}. \end{aligned} \tag{28}$$

Based on them, we adopt *mean Average Precision* (mAP) to evaluate the algorithm effectiveness in our experiment. This metric has been widely used

[5]The complexity of lasso algorithms is $O(nM^2 + M^3)$, but usually, $n \gg M$.

in the literatures including [17], [62] due to its good discriminative power and stability to evaluate the performance of the similarity search. Basically, a large mAP indicates better performance that similar instances have high ranks. More specifically, given a query $\mathbf{x}^*$ and a set of R retrieved instances, the *Average Precision* (AP) is defined as:

$$\mathrm{AP}(\mathbf{x}^*) = \frac{1}{L}\sum_{r=1}^{R} P_r(\mathbf{x}^*) I_r(\mathbf{x}^*),$$

where L is the number of relevant instances in retrieved set; P_r, the precision of top r retrieved instances, refers to the ratio between the number of relevant instance retrieved and the number of retrieved instance r. I_r is an indicator function, which is equal to 1 if the r-th retrieved instance is relevant or 0 otherwise. The APs for all queries are averaged to obtain mAP.

In addition to mAP, we also use Recall-N to measure the similarity-preserving as suggested in [17] on SIFT1M [59] dataset. Let $S_k^d(\mathbf{x}, \Omega)$ be k-nearest neighbors of $\mathbf{x}$ in space Ω using metric d, and let d_E and d_H denote the Euclidean and Hamming distance metric respectively. For example, given a query $\mathbf{x}^*$, $S_{10}^{d_E}(\mathbf{x}^*, \mathbf{X})$ denotes the top 10 nearest neighbors of that query in Euclidean space. $S_{10}^{d_E}(\mathbf{x}^*, \mathbf{X})$ is obtained by a brute force search and is regarded as the ground truth in our experiment. Therefore, Recall-N($\mathbf{x}^*$) is computed by:

$$\text{Recall-}N(\mathbf{x}^*) = \frac{|S_N^{d_H}(h(\mathbf{x}^*), h(\mathbf{X})) \cap S_{10}^{d_E}(\mathbf{x}^*, \mathbf{X})|}{10},$$

where $S_N^{d_H}(h(\mathbf{x}^*), h(\mathbf{X}))$ denotes the query's N nearest neighbors in Hamming space. Recall-N is obtained by averaging the Recall-$N(\cdot)$ over all queries.

Moreover, we also report two additional types of performance curves that are used in the prior arts. One is the *precision-recall* curve showing the precision at different recall level, and the other one is *topN-precision* curve reflecting the change of precision with respect to the number of retrieved instances.

4.1.2. Implementation Details

We first apply PCA technique to reduce the feature dimension to 64. which can also alleviate the influence of noise. Afterwards, the length of sparse codes,

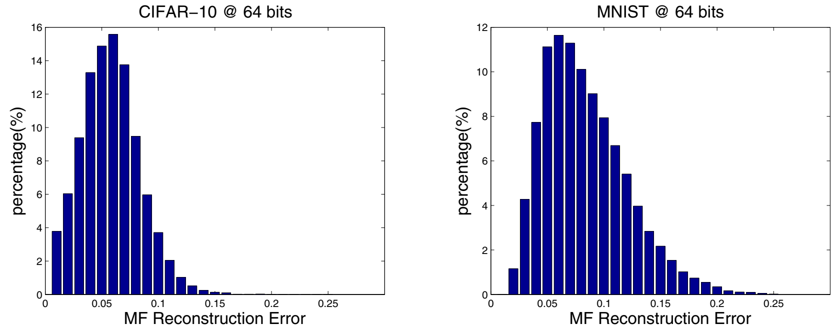


Figure 2: The reconstruction error of matrix factorization.

i.e., the size of dictionary $\mathbf{B}$, is set to 512, and the sparse parameter λ is set to 0.2. SEH has two model parameters: μ and γ. The former controls the compression of sparse coding while the latter determines the similarity-preserving of compressed features. When comparing SEH with the baseline methods, we fix μ and γ to be 1 in all experiments. For the baseline methods, we perform a grid search to tune their parameters and report the best results. Moreover, we set $R = 100$, and all the results are averaged over 10 runs to remove any randomness.

4.2. Similarity-Preserving Task

4.2.1. Reconstruction Error of Matrix Factorization

We investigate the reconstruction error of inequalities in Eq. (17) based on two public datasets. The first dataset is CIFAR-10 [63], in which 60,000 images have been manually grouped into 10 ground-truth classes. Each image is represented by a 512-dimension GIST [64] descriptor and is assigned to one class. The second dataset is MNIST[6], which is made up of 70,000 hand-written digits from 0 to 9. Each image in this dataset is represented by a 784-dimension feature with gray-scale values. We randomly select 10,000 pairs to draw the reconstruction error histogram. In order to eliminate the influence caused by different data dimensions, the original features are normalized, i.e. we have

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

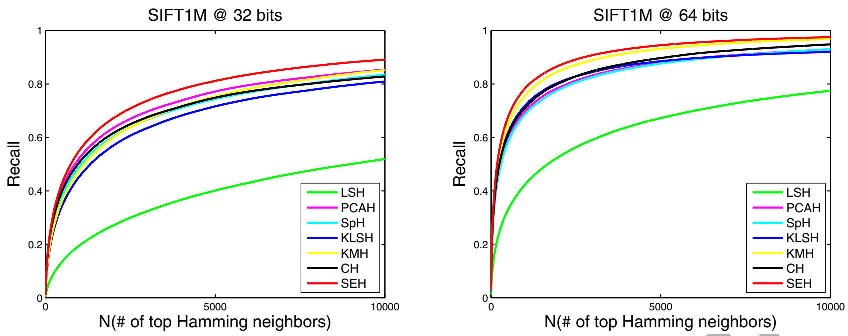


Figure 3: The recall curve on SIFT1M dataset

$\|\mathbf{x}_i\| = 1$. The statistical distribution is shown in Fig. 2. As can be seen, more than 95% reconstruction errors fall into the range of $[0, 0.2]$, which means that reconstruction errors of MF are indeed small, and the bounds in inequalities (Eq.17) are tight.

4.2.2. Euclidean Similarity-Preserving

Euclidean Similarity-preserving requires that the hash methods should map features that are close in Euclidean space to the binary codes that are similar in Hamming space. Here, Recall-N suggested by [17] is measured based on SIFT1M [59] dataset, which contains 1 million 128-dimension SIFT features and $10,000$ independent queries. To highlight the superiority of our algorithm, we compare it with the following state-of-the-art unsupervised hashing methods:

- Locality Sensitive Hashing [7] (LSH)[7],
- PCA Hashing [16] (PCAH)[7],
- Spectral Hashing [12] (SpH)[8],
- Kernelized Locality-sensitive Hashing[8](KLSH)[8],
- K-means Hashing[17](KMH)[8],

[7]We implemented it ourselves because the code is not publicly available.
[8]The source code is kindly provided by the authors.

Table 1: Single-modal retrieval mAP comparison on three datasets.

Task	CIFAR-10			MNIST			NUS-WIDE		
	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
LSH	0.1492	0.1841	0.2181	0.3821	0.5826	0.7018	0.3982	0.4589	0.4732
PCAH	0.2273	0.2439	0.2442	0.6890	0.7710	0.7813	0.4400	0.4761	0.4668
SpH	0.2152	0.2280	0.2405	0.6887	0.7759	0.8057	0.3712	0.4096	0.4626
KLSH	0.1781	0.1830	0.2094	0.5826	0.7484	0.7869	0.3631	0.4216	0.4435
KMH	0.2747	0.2756	0.3037	0.7348	0.8101	0.8228	0.4325	0.5012	0.5054
CH	0.2496	0.2686	0.2984	0.5659	0.8022	0.8234	0.4171	0.4642	0.4934
SEH	**0.2956**	**0.3288**	**0.3619**	**0.8038**	**0.8969**	**0.9157**	**0.5015**	**0.5434**	**0.5523**

- Compressed Hashing[28](CH)[7].

The curves shown on Fig. 3 reveal that our method consistently outperforms all the other competitors when required bit number is varying. It can be observed that LSH is far behind of the other approaches in terms of the performance, because it is a data-independent method. PCAH performs well in the case of 32 bits hash, but it is inferior when a long-bit code is required. The reason might be that very low-variance directions will be picked up as the increased code length [58]. KMH, an affinity-preserving quantization method, performs very well with 64 bits, but it has a significant performance drop when a short code length is required.

4.3. Single-modal Retrieval Task

We evaluate the performance of conducting single-modal retrieval task on CIFAR-10, MNIST and NUS-WIDE[9]. CIFAR-10 is described in section 4.2.1, and 50, 000 images are selected as the database and the rest forms the query set. Images are considered to be relevant if they share the same label. Similarly, 60, 000 images from MNIST are chosen as the database and the rest are supposed to be the query set. Images are considered to be relevant only if they are the same digit. NUS-WIDE [65] contains 10 concepts and each image is adhered to at least

[9]http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

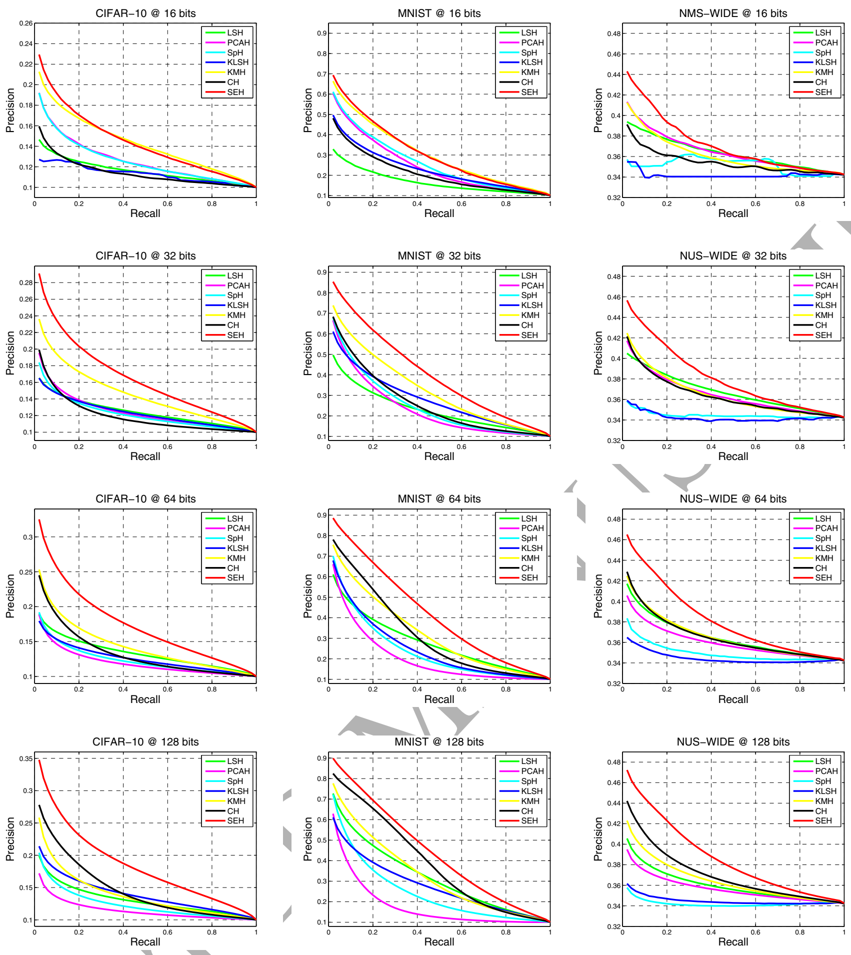


Figure 4: PR-curves of conducting single-modal retrieval task on CIFAR-10, MNIST and NUS-WIDE with different code lengths.

one of them. Each image is represented by a 500-dimension SIFT histogram. We select 5,000 images as the query set and the remaining constitutes the database. Images are assumed to be relevant if they share at least one concepts.

Unlike the test of preserving the similarity, Single-modal Retrieval Task is used to verify the capability of retrieving semantically related results. Again, we compare our SEH with LSH, PCAH, SpH, KLSH, KMH and CH. The mAP values achieved by different approaches are listed in Table 1 and the corresponding PR curves are shown in Fig. 4. Again, our algorithm consistently performs the best over three test datasets, though some methods such as KMH and CH are pretty close to our algorithm at certain situations with respect to the performance. To some extent, the results reflect the property of the algorithm. For instance, the performance of Spectral hashing drops when the code length increases to 128. This is due to the fact that it uses eigenvalue decomposition on affinity matrix to learn hash functions, leading to orthogonality constraints.

4.4. Cross-modal Retrieval Task

As we mentioned before, SEH is able to handle the cross-modal retrieval. To test it, we conduct experiments on three commonly used real-world datasets. The first dataset is Wiki[10], which is a collection of 2,866 Wikipedia multimedia documents. Each document contains 1 image and at least 70 words, where the image is represented by a 128-dimension SIFT histogram and the text is represented by a 10-dimension topic vector generated by LDA model [66]. Totally 10 categories are included in this dataset and each document (image-text pair) is labeled by one of them. The second dataset is LabelMe[11], which is made up of 2688 images. Each image is annotated by several tags depending on the objects in this image. Tags occurred in less than 3 images are discarded and eventually 245 unique tags are remained. This dataset is divided into 8 unique outdoor scenes with the constraint that each image belongs to one scene. The image is represented by a 512-dimension GIST [64] feature and the text is represented by an index vector of selected tags. The last dataset is NUS-WIDE, which is already introduced before. Note that all these three datasets consist of text and images, and we alternately use text and image as queries to search their semantically counterparts in this cross-modal retrieval task. Pairs of image and text are considered to be relevant if they share at least one same concept.

SEH(LSSH)[12] is compared with the following state-of-the-art cross-modal

[10] http://www.svcl.ucsd.edu/projects/crossmodal/

[11] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[12] It is worth mentioning that Latent Semantic Sparse Hashing (LSSH) [24], published on

Table 2: Cross-modal retrieval mAP comparison on three datasets.

	Method	Wiki			LabelMe			NUS-WIDE		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Img to Txt	CVH	0.1984	0.1490	0.1182	0.4704	0.3694	0.2667	0.4694	0.4656	0.4705
	IMH	0.1922	0.1760	0.1572	0.3593	0.2865	0.2414	0.4564	0.4566	0.4589
	DFH	0.2097	0.1995	0.1943	0.4994	0.4213	0.3511	0.4774	0.4677	0.4674
	CHMIS	0.1942	0.1852	0.1796	0.4894	0.4010	0.3414	0.3596	0.3652	0.3565
	SEH	**0.2330**	**0.2340**	**0.2387**	**0.6692**	**0.7109**	**0.7231**	**0.4933**	**0.5006**	**0.5069**
Txt to Img	CVH	0.2590	0.2042	0.1438	0.5778	0.4403	0.3174	0.4800	0.4688	0.4636
	IMH	0.3717	0.3319	0.2877	0.4346	0.3323	0.2771	0.4600	0.4581	0.4653
	DFH	0.2692	0.2575	0.2524	0.5800	0.4310	0.3200	0.5174	0.5077	0.4974
	CHMIS	0.1942	0.1852	0.1796	0.4894	0.4010	0.3414	0.3596	0.3652	0.3565
	SEH	**0.5571**	**0.5743**	**0.5710**	**0.6790**	**0.7004**	**0.7097**	**0.6250**	**0.6578**	**0.6823**

hash methods, which include:

- Cross-view Hashing[19](CVH)[7] ,
- Data Fusion Hashing [21] (DFH)[8] ,
- Inter-media Hashing [23](IMH)[7] ,
- Composite Hashing with Multiple Information Sources [18] (CHMIS)[8] .

The mAPs achieved by different methods are shown in Table 2, and their corresponding performance curves are presented in Fig. 5 and Fig. 6. It can be seen that SEH significantly outperforms all baseline methods on both cross-modal similarity search tasks. When closely looking at the results, it is noticed that the semantic gap between two views of Wiki is quite large. In this case, it seems that the text has better capability to describe the topic than the image. This potentially interprets why the performance becomes much better when the query is a text, compared to the case if the query is an image. Additionally, SEH can reduce the semantic gap between modalities in database since the relevant text and image share the same hash codes (same as CHMIS). That is why SEH can improve mAP by 18%, compared to the best baseline algorithm.

SIGIR, is the cross-modal retrieval version of our proposed framework.

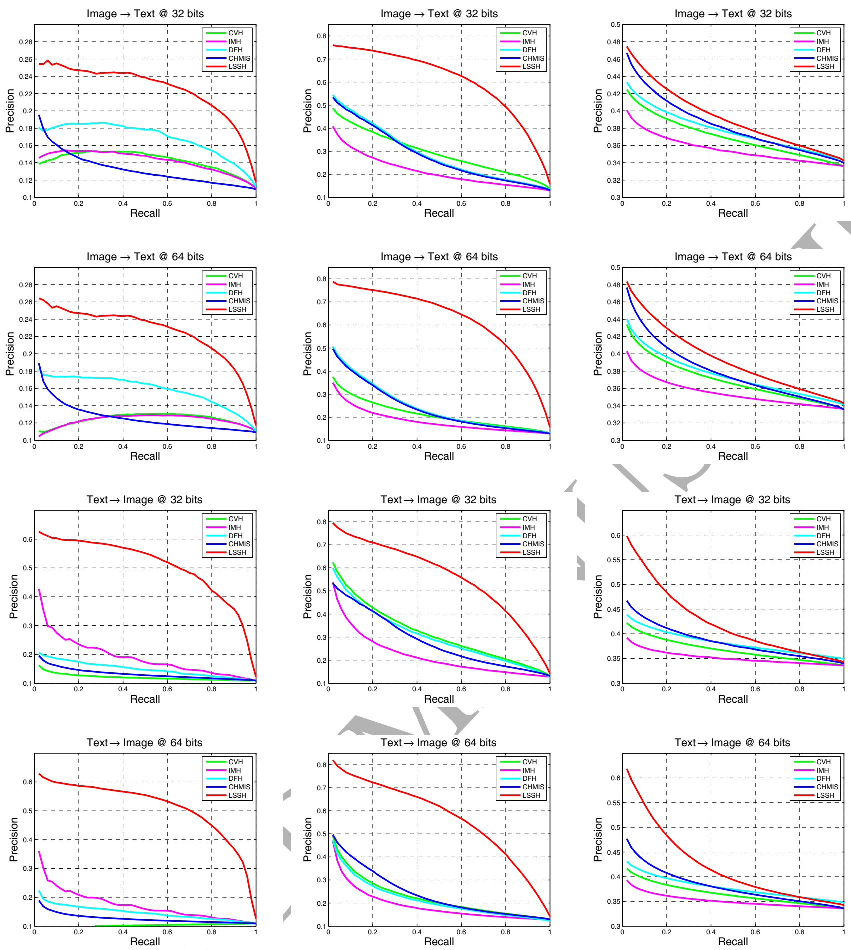


Figure 5: PR-curves of conducting cross-modal retrieval task on Wiki(Left), LabelMe(Middle) and NUS-WIDE(Right) with different code lengths.

It is worth pointing out that the PR curves of several methods look irregular. For example, the PR curve of CVH when querying from text to image at 64 bits shows that it behaves like a random guess. This phenomenon was also reported in [62] and [22]. A reasonable explanation given by [16] is the hash codes will be dominated by bits with very low-variance as the increased code length. Consequently, these indiscriminative hash bits may force the method to make a random guess. However, SEH performs better even for longer length of

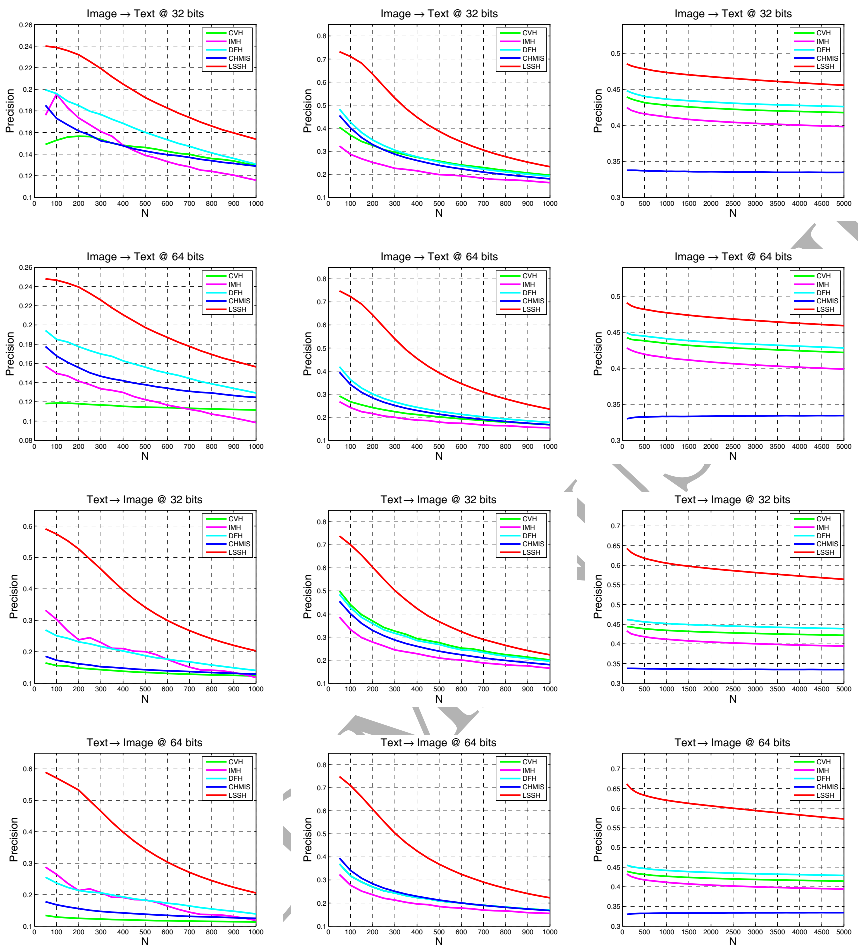


Figure 6: TopN-precision-curves of conducting cross-modal retrieval task on Wiki(Left), LabelMe(Middle) and NUS-WIDE(Right) with different code lengths.

hash codes because SEH can learn more precise descriptions with more latent concepts.

4.5. Parameter Sensitivity Analysis

Moreover, we conduct an empirical analysis on parameter sensitivity over all datasets, because it is important to know how the algorithm behaviors when changing the parameters. Our idea is that we keep the other parameters fixed to

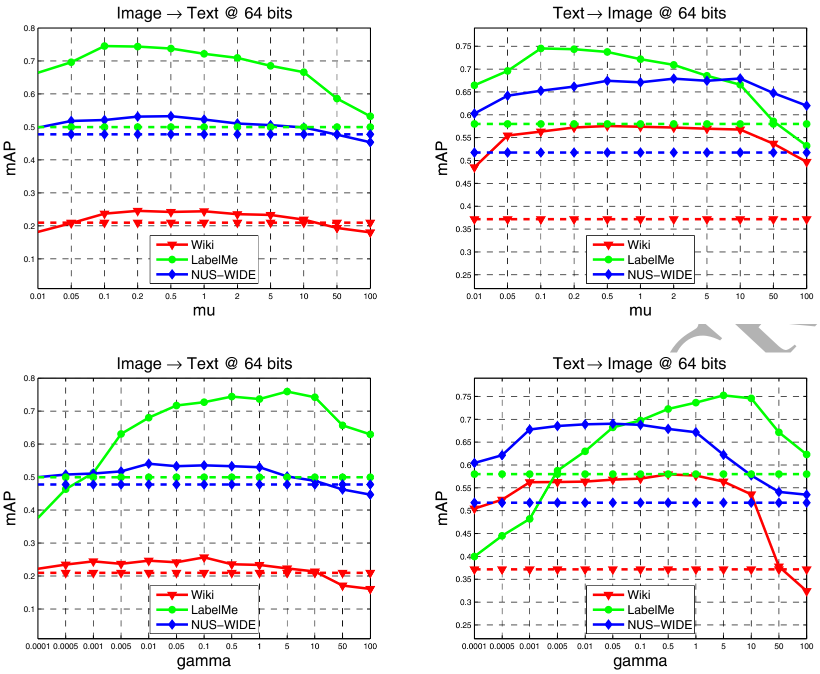


Figure 7: Parameter sensitivity analysis

the settings mentioned in section 4.1.2 when analyzing one particular parameter. Due to limited space, we only present the results at 64 bits on all datasets in Fig. 7. The dashed lines are the best performance of baselines with all experiment settings. For instance, the red dashed line in the first figure shows the result of DFH at 16 bits, which, as be observed from Tab. 2, is the best result of all baselines varying code length for 'Image to Text' task.

The parameter μ leverages the power of images and texts. Actually, utilizing the information from both modals can lead to better results. When μ is too small, e.g., $\mu < 0.05$, our model just focuses on images while ignoring texts. When μ is too large, e.g., $\mu > 10$, our model prefers information from texts. Specifically, it is easy to choose a proper value for μ because we can observe that SEH shows stable and superior performance when $\mu \in [0.05, 10]$.

The parameter γ controls the connection of latent semantic spaces. If γ

is too small, the connection between different modals is weak with imprecise projection in Eq. (18), which will lead to poor performance for cross-modal similarity search. However, if γ is too large, the strong connection twill make the learning of latent representations of images and texts, i.e., Sparse Coding and Matrix Factorization, to be quite imprecise. Because images and texts are represented by imprecise features, it is reasonable that the performance will degrade. Fortunately, it is also effortless to choose proper γ from the range [0.005, 10].

5. Conclusion

In this paper, we have proposed a Sparse Embedded Hashing technique, which is inspired by the excellent capability of sparse coding for image representation. The major difference between traditional algorithms and our algorithm lies in the fact that we implement the sparse representing, affinity preserving and linear embedding in one objective function. Moreover, matrix factorization technique is employed to preserve visual or text (if available) global similarity structure of the original data points. The flexibility of this technique enables us to handle single-modal retrieval and cross-modal retrieval in one system. Extensive evaluations on both single- and cross-modal retrieval tasks reveal that our SEH provides significant advantages over state-of-the-art hashing methods for CBIR.

6. Acknowledgement

This research was supported by the National Natural Science Foundation of China Grant No. 61571269 and 61271394, and the Royal Society Newton Mobility Grant IE150997.

References

[1] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: ACM Symposium on Theory of Computing, ACM, 1998, pp. 604–613.

[2] A. Gionis, P. Indyk, R. Motwani, et al., Similarity search in high dimensions via hashing, in: International Conference on Very Large Data Bases, 1999, pp. 518–529.

[3] D. Wang, X. Gao, X. Wang, Semi-supervised constraints preserving hashing, Neurocomputing 167 (2015) 230–242.

[4] G. Ding, Y. Guo, J. Zhou, Y. Gao, Large-scale cross-modality search via collective matrix factorization hashing, IEEE Transactions on Image Processing 25 (11) (2016) 5427–5440.

[5] Z. Lin, G. Ding, J. Han, J. Wang, Cross-view retrieval via probability-based semantics-preserving hashing, IEEE Transactions on Cybernetics.

[6] D. Zhang, J. Wang, D. Cai, J. Lu, Self-taught hashing for fast similarity search, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2010, pp. 18–25.

[7] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2006, pp. 459–468.

[8] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: IEEE International Conference on Computer Vision, IEEE, 2009, pp. 2130–2137.

[9] M. Kan, D. Xu, S. Shan, X. Chen, Semisupervised hashing via kernel hyperplane learning for scalable image search, IEEE Transactions on Circuits and Systems for Video Technology (2014) 704–713.

[10] G. Shakhnarovich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, in: IEEE International Conference on Computer Vision, IEEE, 2003, pp. 750–757.

[11] R. Salakhutdinov, G. Hinton, Semantic hashing, International Journal of Approximate Reasoning 50 (7) (2009) 969–978.

[12] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.

[13] Z. Bodó, L. Csató, Linear spectral hashing, Neurocomputing 141 (2014) 117–123.

[14] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2074–2081.

[15] C. Strecha, A. Bronstein, M. Bronstein, P. Fua, Ldahash: Improved matching with smaller descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (1) (2012) 66–78.

[16] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for scalable image retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3424–3431.

[17] K. He, F. Wen, J. Sun, K-means hashing: an affinity-preserving quantization method for learning binary compact codes, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 2938–2945.

[18] D. Zhang, F. Wang, L. Si, Composite hashing with multiple information sources, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 225–234.

[19] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: International Joint Conference on Artificial Intelligence, AAAI Press, 2011, pp. 1360–1365.

[20] S. Kim, Y. Kang, S. Choi, Sequential spectral learning to hash with multiple representations, in: European Conference on Computer Vision, Springer, 2012, pp. 538–551.

[21] M. M. Bronstein, A. M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3594–3601.

[22] Y. Zhen, D.-Y. Yeung, Co-regularized hashing for multimodal data, in: Advances in Neural Information Processing Systems, 2012, pp. 1385–1393.

[23] J. Song, Y. Yang, Y. Yang, Z. Huang, H. T. Shen, Inter-media hashing for large-scale retrieval from heterogeneous data sources, in: ACM SIGMOD International Conference on Management of Data, ACM, 2013, pp. 785–796.

[24] J. Zhou, G. Ding, Y. Guo, Latent semantic sparse hashing for cross-modal similarity search, in: ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2014, pp. 415–424.

[25] J. Masci, M. M. Bronstein, A. M. Bronstein, J. Schmidhuber, Multimodal similarity-preserving hashing, IEEE transactions on pattern analysis and machine intelligence 36 (4) (2014) 824–830.

[26] A. Cherian, V. Morellas, N. Papanikolopoulos, Robust sparse hashing, in: IEEE International Conference on Image Processing, IEEE, 2012, pp. 2417–2420.

[27] X. Zhu, Z. Huang, H. Cheng, J. Cui, H. T. Shen, Sparse hashing for fast multimedia search, ACM Transactions on Information Systems 31 (2) (2013) 9.

[28] Y. Lin, R. Jin, D. Cai, S. Yan, X. Li, Compressed hashing, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 446–451.

[29] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, Y. Zhuang, Sparse multi-modal hashing, IEEE Transactions on Multimedia 16 (2) (2014) 427–439.

[30] A. Cherian, Nearest neighbors using compact sparse codes, in: International Conference on Machine Learning, 2014, pp. 1053–1061.

[31] Y. Han, F. Wu, D. Tao, J. Shao, Y. Zhuang, J. Jiang, Sparse unsupervised dimensionality reduction for multiple view data, IEEE Transactions on Circuits and Systems for Video Technology (2012) 1485–1496.

[32] Y. Guo, G. Ding, L. Liu, J. Han, L. Shao, Learning to hash with optimized anchor embedding for scalable retrieval, IEEE Transactions on Image Processing.

[33] B. A. Olshausen, et al., Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature (1996) 607–609.

[34] S. Zhao, H. Yao, X. Jiang, X. Sun, Predicting discrete probability distribution of image emotions, in: IEEE International Conference on Image Processing, IEEE, 2015, pp. 2459–2463.

[35] S. Zhao, H. Yao, X. Jiang, Predicting continuous probability distribution of image emotions in valence-arousal space, in: ACM International Conference on Multimedia, ACM, 2015, pp. 879–882.

[36] S. Zhao, H. Yao, Y. Gao, R. Ji, G. Ding, Continuous probability distribution prediction of image emotions via multi-task shared sparse regression, IEEE Transactions on Multimedia.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3360–3367.

[38] Z. Lu, Y. Peng, Latent semantic learning by efficient sparse coding with hypergraph regularization., in: AAAI Conference on Artificial Intelligence, 2011.

[39] Z. Lin, G. Ding, M. Hu, Y. Lin, S. S. Ge, Image tag completion via dual-view linear sparse reconstructions, Computer Vision and Image Understanding 124 (2014) 42–60.

[40] B. A. Olshausen, D. J. Field, Sparse coding with an overcomplete basis set: A strategy employed by v1?, Vision Research 37 (23) (1997) 3311–3325.

[41] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1794–1801.

[42] M. Yang, L. Zhang, J. Yang, D. Zhang, Robust sparse coding for face recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 625–632.

[43] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing 15 (12) (2006) 3736–3745.

[44] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Transactions on image processing 17 (1) (2008) 53–69.

[45] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, in: Advances in Neural Information Processing Systems, 2006, pp. 801–808.

[46] A. Y. Ng, Feature selection, l 1 vs. l 2 regularization, and rotational invariance, in: International Conference on Machine Learning, ACM, 2004, p. 78.

[47] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Advances in Neural Information Processing Systems, 2009, pp. 2223–2231.

[48] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Transactions on Image Processing 24 (12) (2015) 5659–5670.

[49] S. Gao, I. W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely–laplacian sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3555–3561.

[50] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, IEEE Transactions on Image Processing 20 (5) (2011) 1327–1336.

[51] T. Ge, Q. Ke, J. Sun, Sparse-coded features for image retrieval, in: British Machine Vision Conference, 2013.

[52] R. Ye, X. Li, Compact structure hashing via sparse and similarity preserving embedding, IEEE transactions on cybernetics 46 (3) (2016) 718–729.

[53] A. Z. Broder, On the resemblance and containment of documents, in: International Conference on Compression and Complexity of Sequences, IEEE, 1997, pp. 21–29.

[54] W. Liu, J. Wang, S. Kumar, S. F. Chang, Hashing with graphs, in: International Conference on Machine Learning, 2011, pp. 1–8.

[55] Y. Gong, S. Lazebnik, Iterative quantization: A procrustean approach to learning binary codes, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 817–824.

[56] Y. Guo, G. Ding, J. Han, X. Jin, Robust iterative quantization for efficient ℓ_p-norm similarity search, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 3382–3388.

[57] W. Kong, W.-J. Li, Double-bit quantization for hashing, in: AAAI Conference on Artificial Intelligence, 2012.

[58] K. I. Kim, F. Steinke, M. Hein, Semi-supervised regression using hessian energy with an application to semi-supervised dimensionality reduction, in: Advances in Neural Information Processing Systems, 2009, pp. 979–987.

[59] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE transactions on pattern analysis and machine intelligence 33 (1) (2011) 117–128.

[60] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American society for information science 41 (6) (1990) 391.

[61] Y. Guo, G. Ding, J. Zhou, Q. Liu, Robust and discriminative concept factorization for image representation, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 115–122.

[62] Y. Zhen, D. Yang, A probabilistic model for multimodal hash function learning, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2012, pp. 940–948.

[63] A. Krizhevsky, Learning multiple layers of features from tiny images, in: Tech Report. University of Toronto, 2009.

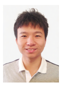
[64] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.

[65] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from national university of singapore, in: ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 48.

[66] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (Jan) (2003) 993–1022.

Biography

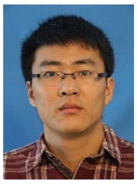
Guiguang Ding received his Ph.D degree in electronic engineering from the University of Xidian. He is currently an associate professor of School of Software, Tsinghua University. Before joining school of software in 2006, he worked as a postdoctoral researcher in automation department of Tsinghua University. His current research centers on the area of multimedia information retrieval and mining, in particular, visual object classification, automatic semantic annotation, content-based multimedia indexing, and personal recommendation. He has published about 40 research papers in international conferences and journals and applied for 18 Patent Rights in China.



Jile Zhou received the B.S. degree in mathematics from Jilin University, Jilin, China, in 2011. He is currently pursuing the M.S. degree at the School of Software, Tsinghua University, Beijing, China. His research interests include multimedia content analysis, indexing and retrieval, and machine learning.

Yuchen Guo received his B.Sc. degree from School of Software, and B.Ec from School of Economics and Management, Tsinghua University, Beijing, China in 2013, and currently is a Ph.D. candidate in School of Software in the same campus. His research interests include multimedia data management, machine learning and data mining.

Zijia Lin received his B.Sc. degree from School of Software, Tsinghua University, Beijing, China in 2011, and currently is a Ph.D. candidate in Department of Computer Science and Technology in the same campus. His research interests include multimedia information retrieval and machine learning.



Sicheng Zhao received the Ph.D. degree from Harbin Institute of Technology in 2016. He is now a postdoctoral research fellow in the School of Software, Tsinghua University, China. His research interests include affective computing, social media analysis and multimedia information retrieval.

Jungong Han is currently a Senior Lecturer with the Department of Computer Science and Digital Technologies at Northumbria University, Newcastle, UK. He received his Ph.D. degree in Telecommunication and Information System from Xidian University, China.% During his Ph.D study, he spent one year at Internet Media group of Microsoft Research Asia, China. Previously, he was a Senior Scientist (2012-2015) with Civolution Technology (a combining synergy of Philips Content Identification and Thomson STS), a %Research Staff (2010-2012) with the Centre for Mathematics and Computer Science (CWI), and a Senior Researcher (2005-2010) with the Technical University of Eindhoven (TU/e) in Netherlands. Dr. Hans research interests include multimedia content %identification, multi-sensor data fusion, computer vision and multimedia security. He has written and co-authored over 80 papers. He is an associate editor of Elsevier Neurocomputing and an editorial board member of Springer Multimedia Tools and Applications. %He has edited one book and organized several special issues for journals such as IEEE T-NNLS and IEEE T-CYB.