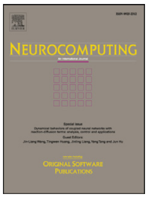
Neurocomputing 253 (2017) 89–96

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Kernel based latent semantic sparse hashing for large-scale retrieval from heterogeneous data sources

Xiangpeng Li[a], Lianli Gao[a,*], Xing Xu[a], Jie Shao[a], Fumin Shen[c], Jingkuan Song[b]

[a] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*
[b] *School of Engineering and Applied Science, University of Columbia, USA*
[c] *School of Computer Science and Technology of China*

ARTICLE INFO

Article history:
Received 10 May 2016
Revised 11 October 2016
Accepted 30 November 2016
Available online 8 March 2017

Keywords:
Hashing
Cross-modal retrieval
Sparse coding
Matrix factorization
Kernel method

ABSTRACT

Recent years, we have witnessed the growing popularity of integrating nearest neighbor search with hashing for effective and efficient similarity search. However, most of the previous cross-modal hashing methods didn't consider the semantic correlation between multi-modal representations and directly project the heterogeneous data into a joint space using a linear projection. To address these challenges and bridge the semantic gap more efficiently. We proposed a method named kernel based latent semantic sparse hashing (KLSSH) in this paper. We firstly capture high-level latent semantic information and then use the equivalence between optimizing the code inner products and the Hamming distances. More specifically, KLSSH firstly employs sparse coding for obtaining primary latent features of image and matrix factorization for generating features of text concepts to learn latent semantic features in a high level abstraction space. Next, it maps the latent semantic feature to compact binary codes using kernel method. Kernel scheme ensures to sequentially and efficiently train the hash functions one bit at a time and then generate very short and discriminative hash codes. Moreover, it reduces the quantization loss obviously at the same time and makes the retrieval performance better. Experiments conducted on three benchmark multi-modal datasets demonstrate the superiority of our proposed method compared with the state-of-the-art techniques.

1. Introduction

Nearest neighbor search based hashing methods play an fundamental role in many applications such as information retrieval [1,30], data mining [2] and computer vision [3]. With the explosion of data, nearest neighbor search faces a severe challenge, hashing is adopted to support large-scale image search by representing a piece of data as a *k*-bit binary hash code, which benefits of low storage costs and high query speed. Among all the hashing method, locality sensitive hashing(LSH) [4] is proposed by mapping the original data into a Hamming space while preserving their similarity with high probability. Based on the LSH, more effective and efficient hashing methods are proposed such as spectral hashing [5], supervised hashing with kernels [6], kernelized hashing [7], semantic hashing [8], iterative quantization hashing [9],K-means hashing [10] and PCA hashing [11]. However, both LSH and its derivations require longer hash codes to achieve higher precision scores, but longer codes always result in a lower recall [4,31]. This is because original metrics are asymptotically preserved in the Hamming space with a increasing code length, while the collision probability of two codes falling into the same hash bucket decreases exponentially as the code length increases.

In terms of data sources, the existing hashing methods can be divided into two categories: single-view hashing [4] and cross-view hashing method [12]. More specifically, the former one is focusing on processing single view data source, while the latter one aims to conduct multi-modal search across various multimedia data sets, such as text, image and video. Moreover, multi-modal search is used to solve the problem of multi-view retrieval for data with different views (e.g., image, text and video). Recently, various hashing methods for multi-modal search have been proposed, including CMFH [13], IMH [12], LSSH [14], CVH [3], CHMIS [15], and DFH [16]. However, to our knowledge, most of the previous approaches have not bridged the gap between multiple views and considered the loss of information in the final representation due to quantization errors. This may lead to low precision in nearest neighbor search.

In this paper, we propose a cross-view retrieval method named kernel based latent semantic sparse hashing(KLSSH). In particular, KLSSH assumes that each view of one instance generates identical

[*] Corresponding author.
E-mail address: lianli.gao@uestc.edu.cn (L. Gao).

http://dx.doi.org/10.1016/j.neucom.2016.11.081
0925-2312/© 2017 Published by Elsevier B.V.

hash codes. Unlike the previous work, KLSSH not only maps the image feature and text feature into a high level semantic space, but also integrates a kernel function to manipulate the code inner product for optimizing the quantization loss. Moreover, KLSSH adopts sparse coding(SC) to capture the salient structures of image, and matrix factorization to learn the latent features from text. Next the learned latent semantic features are mapped into a joint abstract space. Finally, it integrates a kernel formulation with a hash function to effectively lower the quantization loss and preserves the structure of image and text.

In this paper, we design a novel kernel based latent semantic sparse hashing approach for studying the problem of cross-view retrieval for data with multiple views. The contributions of this paper can be summarized as below:

- We propose an innovative cross-model hashing framework to map multi-view features to a joint abstraction space and integrate kernel with hashing for cross-view retrieval.
- We introduce an iterative strategy to support cross-correlation exploration and the training of hash functions one bit at a time.
- Extensive experiments conducted on three real-world large-scale datasets demonstrate the effectiveness and efficiency of our proposed model compared to state-of-the-art hashing algorithms.

The remainder of this paper is organized as follows. Related work and preliminary are presented in Section 2. The details of our proposed method are presented in Section 3. Extensive experimental results are given in Section 4. Lastly, we draw a conclusion in Section 6.

2. Related work

Multi-modal retrieval refers to the task of searching for multimedia information using a user provided query in type of text,image, or video etc. For example, a user can use text to search related images or use image to search related texts. Recently, various hashing methods have been proposed for cross-view retrieval, including supervised ones and unsupervised ones.

Supervised hashing methods [1,3,16–18] are proposed to further exploit available supervised information like labels or semantic affinities of the training data for improving performance. Bronstein et al. [16] proposed DFH that models the projections from features in each view to hash codes as binary classification problems, and learns them with boosting algorithms. Kumar et al. [3] extended the single-view spectral hashing to the case of multiple-view and proposed a novel method named CVH, which learns hash functions via minimizing the similarity-weighted Hamming distance between hash codes of training data. In addition, a half quadratic optimization based algorithm is proposed by Wang et al. [17] to jointly perform common subspace learning and coupled feature selection from different modalities. It integrated coupled linear regression, L_{21} norm and trace norm regularization terms into a generic framework and achieved the state-of-the-art performance for cross-media retrieval task. A novel method is proposed as a supervised cross-view hash method termed SePH [18] that transforms the semantic affinities of training data into a probability distribution and approximates it with to-be-learnt hash codes in Hamming space via minimizing the KL-divergence.

Different form supervised cross-view hashing methods, unsupervised methods methods [12–14,19,20] generally focus on exploiting the intra-view and inter-view relations of the training data with only features in different views to learn the projections from features to hash codes. Song et al. [12] proposed inter-media hashing(IMH) that introduces inter-view and intra-view consistency to learn linear hash functions for mapping features in different views into a common Hamming space. Ding et al. [13] proposed the CMFH, which learns unified hash codes of instances by collective matrix factorization with a latent factor model from different views. Also, [14] further proposed a latent semantic space based hashing method for cross-modal retrieval, where the techniques of sparse coding and matrix factorization are used to learn semantic features and then the isomorphic semantic features are projected into a common latent space using a linear projection. Zhou et al. [19] proposed KSH-CV to learn kernel hash functions via preserving inter-view similarities under an Adaboost framework.

With the development of deep learning, some multi-modal methods based on convolution neural network are proposed and make great significant achievements. Ngiam et al. [21] and [22] target at learning high-dimensional latent features to perform discriminative classification task. Wang et al. [23] applied autoencoder to perform cross-modality retrieval and make a satisfying performance. Kang et al. [24] proposed a multi-view deep learning scheme to learn hash codes with multiple data representations. Wang et al. [25] proposed a novel deep multi-modal hash method which impose orthogonal regularizer on the weighting matrices of the model to reduce redundant information. On the weighting matrices of 121 the model to reduce redundant information. Wang et.al [32] proposed a comprehensive hash survey, thus more hash information can be found in it.

The proposed KLSSH in this paper is a supervised hashing method. We map the image feature using sparse coding and the text features using matrix factorization into a common space just as [14]. But different from previous work mentioned above, we use kernel method in [6] in quantization process and it can effectively reduce the quantization loss. In this way, the learned hash codes can better preserve the semantic of images and texts and perform significantly compared with the state-of-the-art methods.

3. Proposed approach

In this section, we first introduce the framework of our KLSSH followed by more details of each component.

In our KLSSH framework, we propose a novel perspective to solve multi-model retrieval problem. Specifically, the overall flowchart of the KLSSH framework is illustrated in Fig. 1, which consists of two stages: cross-view latent semantic feature learning and kernel based hashing which takes the outputs of the first stage as inputs. In the first stage, the salient structures of images are captured by sparse coding, and the concepts or latent topics are learned from texts by matrix factorization. Next, both the latent representations of image and text are projected into a common and joint latent abstraction space. In the second stage, it takes the joint semantic features as inputs and maps them into a Hamming space with kernel quantization.

Our goal is to generate rich hashing codes for multi-modal dataset for effective and efficient cross view retrieval. Given a set of instances in dataset $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$, where N is the total number of instances. For each instance $\boldsymbol{o}_i$, it contains an image and a text, thus $\boldsymbol{o}_i = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$. Therefore, for $\boldsymbol{O}$, it contains a set of images represented as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{d_1 \times N}$ and a set of text represented as $\boldsymbol{Y} = \{y_1, y_2, \ldots, y_N\} \in \mathbb{R}^{d_2 \times N}$. In addition, suppose both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are projected into a joint and common latent space P to generate a k-dimensional features. Then, for each k-dimensional feature, we generate a r length hash code for it.

3.1. Learning unified semantic features

Image feature extraction. Sparse coding model is able to generalize large-scale data sets complexly, improve the distinguishing ability of features, reduce the time of feature matching and classification and find an representation of image in which each of the components of the representation is only rarely significantly active.

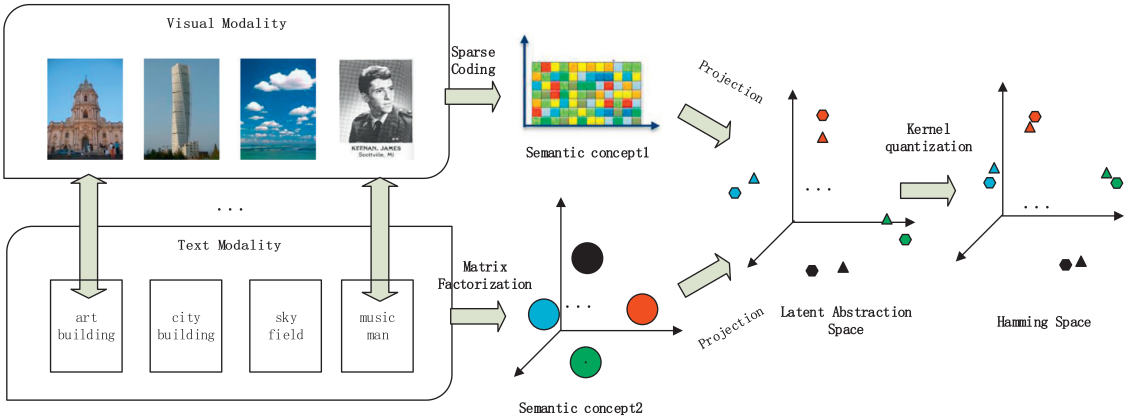


Fig. 1. General framework: KLSSH maps the text and images from their respective natural spaces to two isomorphic latent semantic spaces firstly, then projects the semantic spaces to a joint high level abstract space. Kernel quantization is used to reduce the quantization error finally. The hexagons refer to the semantic form of images and the triangles refer to the semantic form of text in latent abstraction space. We can see clearly the gap between the views of the same instance become small after the kernel quantization.

In the past years, sparse coding has been widely used in extracting better image representation and has proved promising results in [14]. Therefore, we adopt sparse coding to generate the salient information of images in KLSSH. Given $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{d_1 \times N}$, the standard loss with ℓ_1 regularization is:

$$O_{image}(\boldsymbol{B}, \boldsymbol{S}) = \| \boldsymbol{X} - \boldsymbol{BS} \|_F^2 + \sum_{i=1}^{n} \lambda |\boldsymbol{s}_i|_1, \quad (1)$$

where $\boldsymbol{B} \in \mathbb{R}^{d_1 \times M_1}$ is a basis set,and $\lambda > 0$ is the balanced parameter to balance the reconstruction error. $\boldsymbol{S} \in \mathbb{R}^{M_1 \times N}$ denotes semantic features of images after sparse coding.

Text feature extraction and reduction. Matrix factorization is one of the most promising techniques for latent feature extraction and dimensional reduction [14]. Recently, it has been successfully applied in pattern recognition, thus in the paper we apply Matrix Factorization for text feature extraction and reduction. Specifically, given a text $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\} \in \mathbb{R}^{d_2 \times N}$, we use the following matrix factorization function to learn the semantic features:

$$O_{text}(\boldsymbol{U}, \boldsymbol{C}) = \| \boldsymbol{Y} - \boldsymbol{UC} \|_F^2, \quad (2)$$

where $\boldsymbol{U} \in \mathbb{R}^{d_2 \times M_2}$, $\boldsymbol{C} \in \mathbb{R}^{M_2 \times N}$, Column vector $\boldsymbol{U}_{.\,i}$ capture the higher level features of the original text data and $\boldsymbol{C}_{.\,i}$ is the M_2-dimensional representation in latent semantic space.

Learning unified features. Next, we project $\boldsymbol{S} \in \mathbb{R}^{M_1 \times N}$ and $\boldsymbol{C} \in \mathbb{R}^{M_2 \times N}$ into a common latent abstraction space $P \in \mathbb{R}^k$. For each pair of $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, we project their corresponding semantic features $\boldsymbol{s}_i$ and $\boldsymbol{c}_i$ into a same instance $\boldsymbol{p}_i \in P$:

$$\boldsymbol{P}_V \boldsymbol{s}_i = \boldsymbol{p}_i = \boldsymbol{P}_T \boldsymbol{c}_i, \forall i, \quad (3)$$

where $\boldsymbol{P}_V \in \mathbb{R}^{k \times M_1}$ and $\boldsymbol{P}_T \in \mathbb{R}^{k \times M_2}$ are the projection matrices. Similar to the assumption in LSSH [14] that a latent concept can be described by several salient structures of each image. We can rewrite Eq. (3) by left multiplication inverse of $\boldsymbol{P}_T$:

$$\boldsymbol{c}_i = \boldsymbol{P}_T^{-1} \boldsymbol{P}_V \boldsymbol{s}_i = \boldsymbol{P} \boldsymbol{s}_i . \forall i, \quad (4)$$

where $\boldsymbol{P} = \boldsymbol{P}_T^{-1} \boldsymbol{P}_V$ is a linear projection. The Eq. (4) can be approximated to the below formula by optimizing the cross-correlation:

$$\boldsymbol{O}_{cross}(\boldsymbol{P}) = \| \boldsymbol{C} - \boldsymbol{PS} \|_F^2, \quad (5)$$

The overall framework of latent semantic representation object function is

$$\min_{\boldsymbol{B},\boldsymbol{S},\boldsymbol{U},\boldsymbol{C},\boldsymbol{P}} = \boldsymbol{O}_{image} + \theta \boldsymbol{O}_{text} + \mu \boldsymbol{O}_{cross} \quad (6)$$

$$s.t. \; \| \boldsymbol{B}_{.i} \|^2 \leq 1, \| \boldsymbol{U}_{.j} \|^2 \leq 1, \| \boldsymbol{P}_{.t} \|^2 \leq 1, \forall i, j, t,$$

where θ and μ are the parameters that control the discrimination power of images and text latent features, and leverage the liner connection of latent spaces, respectively. The above object function is used to learn the abstraction representations of latent semantic space for both images and texts. After the learning process, the text features are converted to $\boldsymbol{A}_T = \boldsymbol{C} = \{\boldsymbol{a}_{T1}, \ldots, \boldsymbol{a}_{TN}\}$, while the image features are converted to $\boldsymbol{A}_{Ig} = \boldsymbol{PS} = \left\{\boldsymbol{a}_{Ig1}, \ldots, \boldsymbol{a}_{IgN}\right\}$. Therefore, given a dataset $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$, we have the following semantic feature set $\boldsymbol{A} = \left\{\boldsymbol{A}_T, \boldsymbol{A}_{Ig}\right\} = \left\{\boldsymbol{a}_{T1}, \ldots, \boldsymbol{a}_{TN}, \boldsymbol{a}_{Ig1}, \ldots, \boldsymbol{a}_{IgN}\right\}$.

Kernel based latent semantic sparse hashing. Given the latent semantic feature set $\boldsymbol{A} = \left\{\boldsymbol{A}_T, \boldsymbol{A}_{Ig}\right\} = \left\{\boldsymbol{a}_{T1}, \ldots, \boldsymbol{a}_{TN}, \boldsymbol{a}_{Ig1}, \ldots, \boldsymbol{a}_{IgN}\right\}$ including the latent semantic features of both images and texts. We introduce a kernel function $K : R^{2N} \times R^{2N} \longmapsto R$ to construct a hash function for the reason that the kernel trick has been theoretically and empirically proven to be able to tackle practical data that is mostly linearly inseparable.

Suppose the code matrix of labeled data a_i is $\boldsymbol{H}_l$ and l represents the number of labeled data. We can propose a least-squares style objective function Q to learn the codes. Let us write the r-bit hash code of sample a_i as $code_r(a_i) = [h_1(a_i), ..., h_r(a_i)] \in \{1, -1\}^{l \times r}$ and$\|.\|_F$ represents the Frobenius norm.

$$\min_{H_l \in \{1,-1\}^{l \times r}} \boldsymbol{Q} = \| \frac{1}{r} \boldsymbol{H}_l \boldsymbol{H}_l^T - \boldsymbol{W} \|_F^2, \quad (7)$$

where

$$\boldsymbol{H}_l = \begin{bmatrix} code_r(\boldsymbol{a}_1) \\ \cdots \\ code_r(\boldsymbol{a}_l) \end{bmatrix} \quad (8)$$

and $\boldsymbol{W}$ denotes the similarity matrix of labeled data where $\boldsymbol{w}_{i,j} = 1$ if $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are similar, otherwise $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are dissimilar.

Directly optimizing the Hamming distances is nontrivial because of the complex mathematical formula $D_h(\boldsymbol{a}_i, \boldsymbol{a}_j) = |\{k | h_k(\boldsymbol{a}_i) \neq h_k(\boldsymbol{a}_j), 1 \leq k \leq r\}|$. Then we consider the code inner product as below:

$$code_r(\boldsymbol{a}_i) \circ code_r(\boldsymbol{a}_j) = r - 2D_h(\boldsymbol{a}_i, \boldsymbol{a}_j), \quad (9)$$

where $\circ$ stands for the code inner product. The Hamming distance and the code inner product are in one-to-one correspondence, thus it enables an equivalent optimization on code inner products.

We randomly select ν samples uniformly from $\boldsymbol{A}$ and train the kernel function as [26]:

$$f(\boldsymbol{a}) = \sum_{j=1}^{\nu} (\kappa(\boldsymbol{a}_{(j)}, \boldsymbol{a}) - \frac{1}{2N} \sum_{i=1}^{2N} \kappa(\boldsymbol{a}_{(j)}, \boldsymbol{a}_i)) \alpha_j = \alpha^T \bar{k}(\boldsymbol{a}), \quad (10)$$

where $\alpha = [\alpha_1, \dots, \alpha_v]^T$ and $\bar{k} : R^{2N} \longmapsto R^v$ is a vectorial map defined by

$$\bar{k}(\boldsymbol{a}) = [\kappa(\boldsymbol{a}_{(1)}, \boldsymbol{a}) - \mu_1, \dots, \kappa(\boldsymbol{a}_{(h)}, \boldsymbol{a}) - \mu_h]^T \tag{11}$$

in which $\mu_j = \sum_{i=1}^{2N} (\kappa(\boldsymbol{a}_{(j)}, \boldsymbol{a}_i)/2N)$ can be precomputed. α is assigned randomly according to a Gaussian distribution.

As mention above, we can generalize sgn(·) to take the element-wise sign operation for any vectors or matrix inputs, and then express the code matrix $\boldsymbol{H}_l$ as (given $h_k(\boldsymbol{a}) = sgn(\alpha_k^T \bar{k}(\boldsymbol{a})) = sgn(\bar{k}^T(\boldsymbol{a})\alpha_k)$)

$$H_l = \begin{bmatrix} h_1(\boldsymbol{a}_1) & \dots & h_r(\boldsymbol{a}_1) \\ \dots & \dots & \dots \\ h_1(\boldsymbol{a}_l) & \dots & h_r(\boldsymbol{a}_l) \end{bmatrix} = sgn(\bar{K}_l \alpha), \tag{12}$$

where $\bar{K}_l = [\bar{k}(\boldsymbol{a}_1), \dots, \bar{k}(\boldsymbol{a}_l)]^T \in \boldsymbol{R}^{l \times v}$ and $\alpha = [\alpha_1, \dots, \alpha_r] \in \boldsymbol{R}^{v \times r}$.

We substitute $\boldsymbol{H}_l$ in Eq. (7) with (12), then we obtain an analytical form of the objective function Q:

$$\min_{\alpha \in R^{v \times r}} Q(\alpha) = \| \frac{1}{r} sgn(\bar{K}_l \alpha)(sgn(\bar{K}_l \alpha))^T - \boldsymbol{S} \|_F^2 . \tag{13}$$

We use this object function to train our KLSSH model. In the next subsection, we will propose a novel and efficient algorithm to solve this challenging problem.

3.2. Optimization algorithm

We easily find it non-convex of Eq. (6) with five variables $\mathbf{B}, \mathbf{C}, \mathbf{P}, \mathbf{S}, \mathbf{U}$. We can make Eq. (6) to be convexed with the respect to any one of the five variables while fixing the other four. Therefore, the above optimization problem can be solved by an iterative framework.

Updating *S*. By fixing $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{P}$, $\boldsymbol{U}$, the problem of Eq. (6) is equivalent to optimizing the following objective function:

$$\begin{aligned} &\min_{\mathbf{S}} \|\boldsymbol{X} - \boldsymbol{BS}\|_F^2 + \sum_1^N \lambda |\boldsymbol{s}_i|_1 + \theta \|\boldsymbol{C} - \boldsymbol{PS}\|_F^2 \\ &\Leftrightarrow \min_S \left\| \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\theta}\boldsymbol{C} \end{bmatrix} - \begin{bmatrix} \boldsymbol{Y} \\ \sqrt{\theta}\boldsymbol{P} \end{bmatrix} \boldsymbol{S} \right\|_F^2 + \sum_1^N \lambda |\boldsymbol{s}_i|_1 \end{aligned} . \tag{14}$$

This problem can be solved by using sparse learning with efficient projection (SLEP) package.

Updating *C*: By fixing $\boldsymbol{B}$, $\boldsymbol{S}$, $\boldsymbol{P}$, $\boldsymbol{U}$, the problem of Eq. (6) is equivalent to optimizing the following objective function:

$$\boldsymbol{C} = \left(\boldsymbol{U}^T \boldsymbol{U} + \frac{\theta}{\mu} \boldsymbol{I} \right)^{-1} \left(\frac{\theta}{\mu} \boldsymbol{RS} + \boldsymbol{U}^T \boldsymbol{Y} \right), \tag{15}$$

where $\boldsymbol{I}$ is the identity matrix.

Updating *B*: By fixing $\boldsymbol{C}$, $\boldsymbol{S}$, $\boldsymbol{P}$, $\boldsymbol{U}$, the problem of Eq. (6) is equivalent to optimizing the following objective function:

$$\boldsymbol{B} = \boldsymbol{X}\boldsymbol{S}^T (\boldsymbol{S}\boldsymbol{S}^T + \Theta)^{-1}, \tag{16}$$

where Θ is a diagonal matrix.

Updating *P*: By fixing $\boldsymbol{C}$, $\boldsymbol{S}$, $\boldsymbol{B}$, $\boldsymbol{U}$, the problem of Eq. (6) is equivalent to optimizing the following objective function:

$$min_R \parallel \boldsymbol{C} - \boldsymbol{PS} \parallel_F^2 \qquad s.t. \parallel \boldsymbol{P}_{\cdot i} \parallel^2 \leq 1, \forall i, \tag{17}$$

Updating *U*: By fixing $\boldsymbol{C}$, $\boldsymbol{S}$, $\boldsymbol{B}$, $\boldsymbol{P}$, the problem of Eq. (6) is equivalent to optimizing the following objective function:

$$\min_U \parallel Y - UC \parallel_F^2 \qquad s.t. \parallel U_{\cdot i} \parallel^2 \leq 1, \forall i. \tag{18}$$

Updating kernel function: Greedy optimization is used to optimize the kernel component in the process of hash code generation. We find the separable property of code inner products so we can solve the hash function in an incremental mode. The Eq. (13) can be rewrite as:

$$\min_\alpha \parallel \sum_{k=1}^r sgn(\bar{\boldsymbol{K}}_l \alpha_k)(sgn(\bar{\boldsymbol{K}}_l \alpha_k))^T - rS \parallel_F^2, \tag{19}$$

where r vectors (α_k) are separated in the summation, thus we can solve α_k sequentially and it involves solving one vector α_k provided with the previous solved vectors $\alpha_1^*, \alpha_2^*, \dots, \alpha_{k-1}^*$. $\boldsymbol{P}_{k-1} = r\boldsymbol{S} - \sum_{t=1}^{k-1} sgn(\bar{\boldsymbol{K}}_l \alpha_k^*)(sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^T \quad (\boldsymbol{P}_0 = \boldsymbol{pS})$. Then α_k an be calculated by minimizing the following cost function:

$$\begin{aligned} &\parallel sgn(\bar{\boldsymbol{K}}_l \alpha_k^*)(sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^T - P_{k-1} \parallel_F^2 \\ &= ((sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^T sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^2 - \\ &\qquad 2(sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^T P_{k-1} sgn(\bar{\boldsymbol{K}}_l \alpha_k^*) + tr(\boldsymbol{R}_{k-1}^2) \\ &= -2(sgn(\bar{\boldsymbol{K}}_l \alpha_k^*))^T P_{k-1} sgn(\bar{\boldsymbol{K}}_l \alpha_k^*) + const, \end{aligned} \tag{20}$$

Motivated by spectral methods for hashing [5], we apply a spectral relation trick to solve the sign functions. Moreover, we replace $sgn()$ with the sigmoid-shaped function $\varphi(\boldsymbol{x}) = 2/(1 + exp(-\boldsymbol{x})) - 1$ which is sufficient smooth and well approximates $sgn(\boldsymbol{x})$ when $|\boldsymbol{x}| > 6$. The algorithm is summarized as Algorithm 1.

Algorithm 1 Kernel based latent semantic sparse feature learning.

```
Input: Image representation matrix $\boldsymbol{X}$ and text feature matrix $\boldsymbol{Y}$, parameters $\lambda, \mu, \theta$ and the number of hash bit $r$. A pairwise label matrix $\boldsymbol{W} \in \boldsymbol{R}^{2N \times 2N}$ defined on $2N$ samples $A_l = \{\boldsymbol{a}_i\}_{i=1}^l$, a kernel function $\kappa : \boldsymbol{R}^{2N} \times \boldsymbol{R}^{2N} \mapsto P$,the number of support samples($m < l$)
Output:$r$ length hash functions $\{h_k(x) = sgn(\bar{k}^T(c)\alpha_k^*)\}_{k=1}^r$ and $2N$ hash codes $H = \{code_r(\boldsymbol{a}_i)\}_{i=1}^{2N}$.
Training:  We randomly select uniform $m$ samples from$A$, and compute zero-centered $v$-dim kernel vectors $\bar{k}(x_i)(i = 1, \dots, n)$ using the kernel function $\kappa$ according to Eq. (11)
1. Initialize $\boldsymbol{U}, \boldsymbol{C}, \boldsymbol{P}$ and $\boldsymbol{B}$ by random matrices respectively,$\boldsymbol{Y}$ by $\ell_2$ norm.
$\boldsymbol{P}_0 = \boldsymbol{pS}$ and $T_{max}$=500
2. repeat:
3.      Fixing $\boldsymbol{U}$, $\boldsymbol{P}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, updating $\boldsymbol{S}$ as illustratedEq. (14).
4.      Fixing $\boldsymbol{U}$, $\boldsymbol{P}$, $\boldsymbol{B}$ and $\boldsymbol{S}$, updating $\boldsymbol{C}$ as illustratedEq. (15).
5.      Fixing $\boldsymbol{U}$, $\boldsymbol{P}$, $\boldsymbol{S}$ and $\boldsymbol{C}$, updating $\boldsymbol{B}$ as illustratedEq. (16).
6.      Fixing $\boldsymbol{U}$, $\boldsymbol{B}$, $\boldsymbol{S}$ and $\boldsymbol{C}$, updating $\boldsymbol{P}$ by optimizing theEq. (17).
7.      Fixing $\boldsymbol{P}$, $\boldsymbol{B}$, $\boldsymbol{S}$ and $\boldsymbol{C}$, updating $\boldsymbol{U}$ by optimizing theEq. (18)
8. until convergence.
9. for $k$=1,...,$r$ do
solving the generalized eigenvalue problem $\bar{\boldsymbol{K}}_l^T \boldsymbol{P}_{k-1} \bar{\boldsymbol{K}}_l \alpha = \lambda \bar{\boldsymbol{K}}_l^T \bar{\boldsymbol{K}}_l \alpha$, obtaining the largest eigenvector $\alpha_k^0$ such that $(\alpha_k^0)^T \bar{\boldsymbol{K}}_l^T \bar{\boldsymbol{K}}_l \alpha_k^0 = \boldsymbol{l}$, using the gradient descent method to optimize $min_\alpha - (\varphi(\bar{\boldsymbol{K}}_l \alpha))^T \boldsymbol{P}_{k-1} \varphi(\bar{\boldsymbol{K}}_l \alpha)$ with the initial solution $\alpha_k^0$ and $T_{max}$ budget iterations, and achieving $\alpha_k^*$;
$h^0 \leftarrow sgn(\bar{\boldsymbol{K}}_l \alpha_k^0), h^* \leftarrow sgn(\bar{\boldsymbol{K}}_l \alpha_k^*)$;
        if  $(h^0)^T \boldsymbol{P}_{k-1} h^0 > (h^*)^T P_{k-1} h^*$then
           $\alpha_k^* \leftarrow \alpha_k^0, h^* \leftarrow h^0$;
        end if $\boldsymbol{P}_k \leftarrow \boldsymbol{P}_{k-1} - h^*(h^*)^T$;
      end for
Hashing code generation: for $i$=1,...,2N,do
$code_r(\boldsymbol{a}_i) \leftarrow [sgn(\bar{k}^T(\boldsymbol{a}_i)\alpha_1^*), \dots, sgn(\bar{k}^T(\boldsymbol{a}_i)\alpha_r^*)]$in detail.
```

4. Experiment

4.1. Experiment settings

4.1.1. Datasets

In this work, three datasets are used for evaluation.

Table 1
Cross-view retrieval performance of the proposed KLSSH and compared baselines on the Wiki, LabelMe, NUS-WIDE datasets with different hash code lengths (i.e., 16bits, 32bits, 64bits and 128 bit), in terms of mAP.

	Method	Wiki				LabelMe				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Image to Text	CVH	0.1984	0.1490	0.1182	0.1133	0.4704	0.3694	0.2667	0.1915	0.4694	0.4656	0.4705	0.4777
	IMH	0.1922	0.1760	0.1572	0.1351	0.3593	0.2865	0.2414	0.1990	0.4564	0.4566	0.4589	0.4453
	DFH	0.2097	0.1995	0.1943	0.1898	0.4994	0.4213	0.3511	0.2788	0.4774	0.4677	0.4674	0.4703
	CHMIS	0.1942	0.1852	0.1796	0.1671	0.4894	0.4010	0.3414	0.2967	0.3596	0.3652	0.3565	0.3594
	LSSH	**0.2330**	**0.2340**	**0.2387**	**0.2340**	0.6692	0.7109	0.7231	0.7333	0.4933	0.5006	0.5069	0.5084
	KLSSH	0.2256	0.2235	0.2161	0.2059	**0.7207**	**0.7607**	**0.7770**	**0.7837**	**0.5020**	**0.5241**	**0.5336**	**0.5485**
Text to Image	CVH	0.2590	0.2042	0.1438	0.1170	0.5778	0.4403	0.3174	0.2153	0.4800	0.4688	0.4636	0.4709
	IMH	0.3717	0.3319	0.2877	0.2674	0.4346	0.3323	0.2771	0.2258	0.4600	0.4581	0.4653	0.4454
	DFH	0.2692	0.2575	0.2524	0.2540	0.5800	0.4310	0.3200	0.2313	0.5174	0.5077	0.4974	0.4903
	CHMIS	0.1942	0.1852	0.1796	0.1671	0.4894	0.4010	0.3414	0.2967	0.3596	0.3652	0.3565	0.3594
	LSSH	0.5571	0.5743	0.5710	0.5577	0.6790	0.7004	0.7097	0.7140	0.6250	0.6578	0.6823	0.6913
	KLSSH	**0.6175**	**0.6289**	**0.6359**	**0.6438**	**0.8406**	**0.8770**	**0.8916**	**0.8938**	**0.6222**	**0.6814**	**0.7030**	**0.7341**

- **Wiki.** The Wiki [27] dataset consists of 2866 multimedia documents collected from the Wikipedia for cross-modal retrieval. For each document, the image is represented by a 128-dimension Bag-of-Visual-Words SIFT feature and the text component is represented as a 10-dimension topic vector, learned with latent Dirichlet allocation. The whole dataset contains 10 semantic classes and each pair is labeled with one of them. In our experiment, We use 75% of the dataset as the training set, the remaining 25% as the query set.
- **LabelMe.** The LabelMe [28] dataset is created by the MIT computer science and artificial intelligence laboratory(CSAIL) which made up of 2688 digital images with annotations. Each image is annotated by several tags. The whole dataset is divided into 8 semantic categories and each image is represented by a 512-dimension GIST feature and each text is represented by a index vector of selected tags. Image-text pairs are regarded as similar if they share the same scene label. We use 75% of the dataset as the training set, the remaining 25% as the query set.
- **NUS-WIDE.** The NUS-WIDE [29] dataset is a real-world web image database containing 81 concepts and 269,648 images with tags. For each instance, the image view is represented as a 500-dimension Bag-of-Visual-Words SIFT feature vector and the text view as a binary tagging vector w.r.t. the top 1000 most frequent tags. Due to the scarcity of some concepts, we select 10 most common concepts,186,577 images as the new dataset. Pairs are considered to be similar if they share at least one concept. We use 98% of the dataset as the training set, the remaining 2% as the query set.

4.1.2. Algorithm for compassion

Our model can retrieve two media types (i.e., text and image) from heterogeneous data sources. In our experiments, we use text and image as query types respectively to retrieve relevant results from the database composed of three datasets mentioned above. For the evaluation purpose, we design two retrieval tasks: (1) the first task (Image to Text) is to use images as queries to retrieve text documents; (2) the second task (Text to Image) is to use texts as queries to retrieve images.

In order to evaluate our approach, we compare our method with a range of the state-of-the-art approaches, including cross-view hashing(CVH) [3] extending spectral hashing for multi-view retrieval, data fusion hashing(DFH) [16] constructing two groups of linear hash functions for preserving the similarity structure in each individual media type, inter-media hashing(IMH) [12] that incorporates inter-media and intra-media consistency to discover a common Hamming space and uses linear regression with regularization model to learn view-specific hash functions, composite hashing with multiple information sources(CHMIS) [15] combining information from multi sources into integrated hash codes, and lLatent semantic sparse hashing(LSSH) [14] which uses the sparse coding and matrix factorization to project the image and text features into a common semantic space.

4.1.3. Implementation details

Following [14], we carried out the experiments by setting the following the parameters. Specifically, for image dataset, we firstly adopt the PCA to remove the noisy point and reduce the dimension to 64. In the process of sparse coding, we set the dimension of sparse codes as 512 and the sparsity parameter $\lambda = 2$. Moreover, μ leverages the discrimination power between images and texts and is set as 0.5. γ controls the linear connection of latent semantic spaces and is set as 1.0. All these parameters shows a good performance on all three datasets.

4.2. Results

4.2.1. Results on Wiki dataset

We report the results of two tasks on the Wiki dataset and the results are shown in Table 1. From Table 1, we can conclude that our approach outperforms the state-of-the-art methods (CVH, IMH, DFH, CHMIS and LSSH) in mAP for both tasks (image queries text and text queries images). Specifically, the MAP of text-query-image shows the significant performance, while the mAP of image-query-text is better than most of the baseline methods. The reason is that the semantic gap between image and text of Wiki is quite large. The semantic information of image is abundant and can not be expressed well by a short code generated by a brief text description, so the performance of image-query-text is not very outstanding. We guess the limitation of Wiki dataset's scale leads to the unsatisfactory performance of image-query-text on Wiki, especially the text information in Wiki. In this dataset, there is a slight of text information of each image, which cause that the hash codes lacking the semantic information of images. We can find this conclusion comparing this experiment with the other datasets that with abundant text data.

Precision-recall and precision-N curves are shown in Fig. 2. We observed that our method outperforms other baseline methods slightly in image-query-text and shows the superiority in text-query-image. In comparison with LSSH, our method can improve the precision with the increase of recall and N. It is clear that the precision of all the methods decrease with the increase of the Recall, but our method still outperforms all the state-of-the-art approaches.

4.2.2. Results on LabelMe dataset

We report the results of two tasks on the LabelMe dataset and the mAP results are reported in Table 1. From Table 1, we can

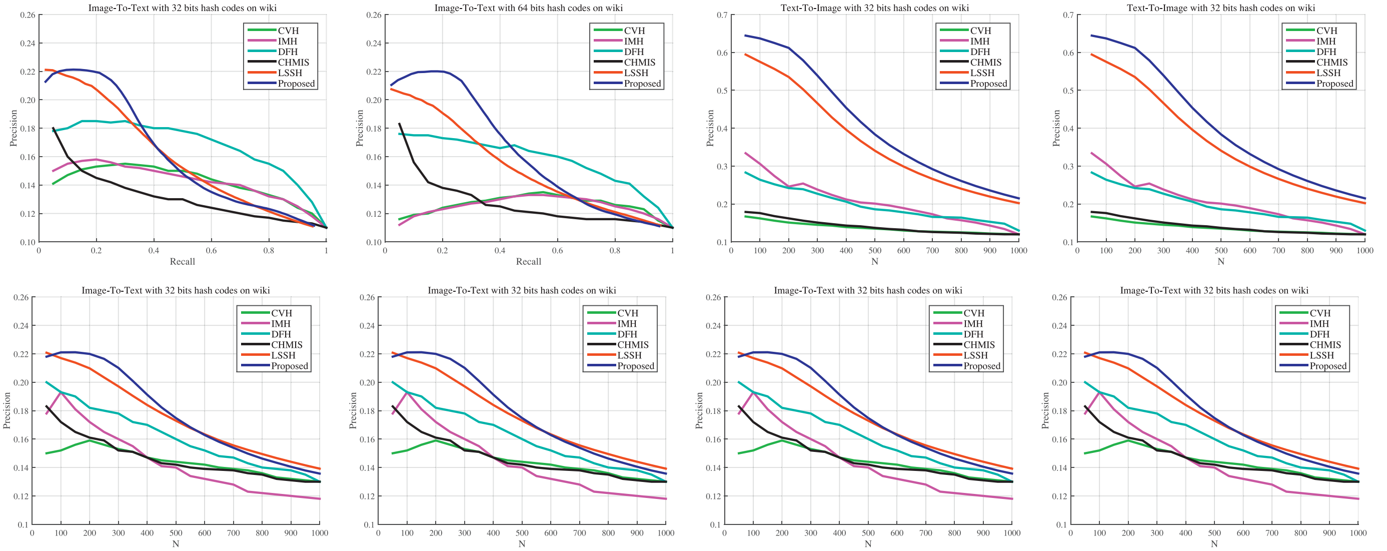


Fig. 2. PR-curves and topN-precision curves on Wiki varying code length.

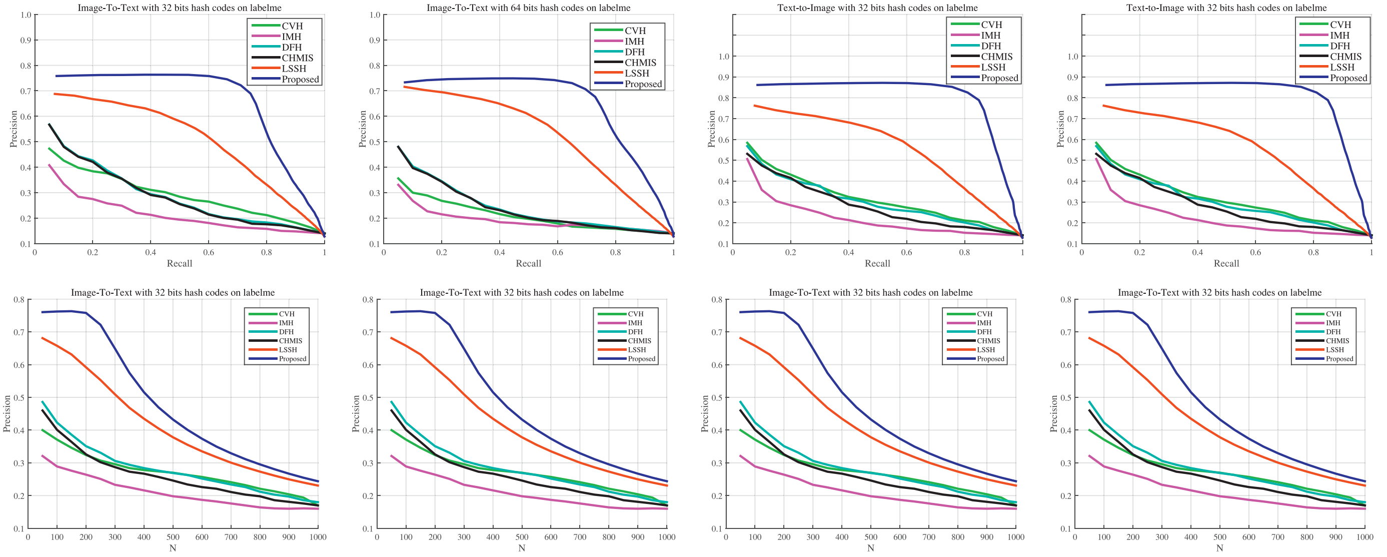


Fig. 3. PR-curves and topN-precision curves on LabelMe varying code length.

observe that KLSSH outperforms state-of-the-art methods on all tasks across different code length. As the code length increases, the mAP performances for all the methods increase too. Similar, the mAP for the text-query-image is actually quite good with (0.8406 of 16 bits, 0.8770 of 32 bits, 0.8916 of 64 bits, 0.8938 of 128 bits) and significantly performs better than LSSH with (0.6790 of 16 bits, 0.7004 of 32 bits, 0.7097 of 64 bits, 0.7382 of 128 bits), which verifies the effectiveness of our method. This is because LSSH directly learns hashing functions rather than using quantization rules. Moreover, our proposed method do not degrade with increasing code length, which is due to we learn more appropriate representation from training instances and encode more useful information with longer codes. Compared to the Wiki dataset, LabelMe has the similar number of instances roughly. But the instance of LabelMe has more text description to the image, and the superiority of KLSSH is performed completely on this dataset.

Precision-recall and precision-N curves are shown in Fig. 3. We observed that our method outperforms other baseline methods significantly for both tasks on LabelMe rather than the performance on Wiki. Due to the abundant text and image information, KLSSH also shows the superiority in PR-curves and topN-precision curves. With the same recall, our method has the higher precision. And when the number of return images or text, we will get the better performance significantly as well.

4.2.3. Results on NUS-WIDE dataset

For the NUS-WIDE dataset, 182, 846 instances are chosen as the database and the remaining 3,731 instances as the query set. Moreover,we select 10,000 instances from databases randomly as the training set to learn hash functions and then they are applied to other instances in database to generate hash codes. The mAP is shown in Table 1, which shows more significant performance than other baseline methods not only short codes but long codes. Our method achieves the best performance with different code lengths. For image-query-text task, our method achieves 0.5020 of 16 bits, 0.5241 of 32 bits, 0.5336 of 64 bits, 0.5485 of 128 bits, while for text-query image task, our method achieves 0.6222 of 16 bits, 0.6814 of 32 bits, 0.7030 of 64 bits, 0.7341 of 128 bits.

Precision-recall and precision-N curves are shown in Fig. 4. We observed that our method outperforms other baseline methods

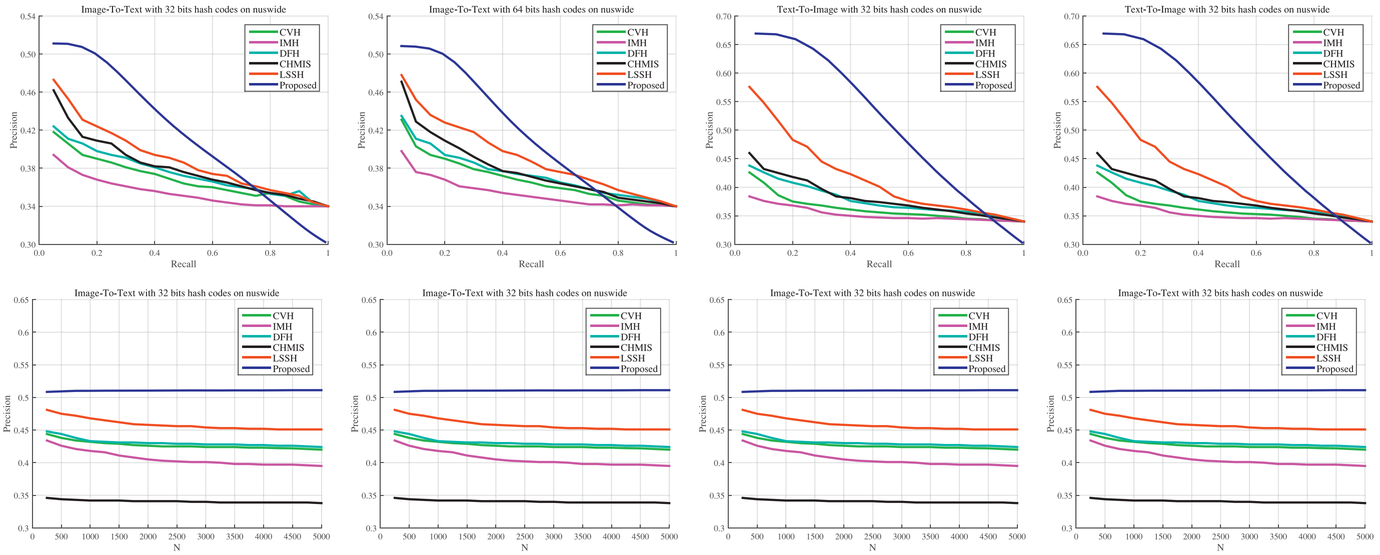


Fig. 4. PR-curves and topN-precision curves on NUS-WIDE varying code length.

significantly for both tasks and it performs better with longer hash codes. Compared to the above two datasets, NUS-WIDE dataset has more images and text descriptions, which leads to the stable performance as showed in Fig. 4. The trend of precision-recall curve and precision-N curve seems more stable and our approach has the obvious superiority to handle large-scale database.

5. Conclusion

In this paper, we have proposed a novel inter-media hashing approach KLSSH to achieve effective and efficient multimedia retrieval from heterogeneous data sources. We firstly bridge the semantic gap more efficiently via capturing high-level latent semantic information and then use the equivalence between optimizing the code inner products and the Hamming distances. Extensive experimental results have shown the superiority of KLSSH over state-of-the-art indexing methods. In future, we plan to integrate deep neural networks with hashing functions to improve the performance of multi-modal retrieval.

Acknowledgment

This work is supported by National Nature Science Foundation of China (No. 61632007 and No. 61502080).

References

[1] X. Xu, F. Shen, Y. Yang, H.T. Shen, Discriminant cross-modal hashing, in: Proceedings of International Conference on Multimedia Retrieval, 2016, pp. 305–308.
[2] L. Gao, J. Song, J. Shao, X. Zhu, H. Shen, Zero-shot image categorization by image correlation exploration, in: Proceedings of the 5th International Conference on Multimedia Retrieval, ACM, 2015, pp. 487–490.
[3] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, 2011, p. 1360.
[4] A. Gionis, P. Indyk, R. Motwani, et al., Similarity search in high dimensions via hashing, in: Proceedings of the 25th International Conference on Very Large Data Bases, vol. 99, 1999, pp. 518–529.
[5] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proceedings of the Advances in Neural Information Processing systems, 2009, pp. 1753–1760.
[6] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2074–2081.
[7] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: Proceedings of the 12th International Conference on Computer Vision, IEEE, 2009, pp. 2130–2137.
[8] R. Salakhutdinov, G. Hinton, Semantic hashing, Int. J. Approx. Reason. 50 (7) (2009) 969–978.
[9] Y. Gong, S. Lazebnik, Iterative quantization: a procrustean approach to learning binary codes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 817–824.
[10] K. He, F. Wen, J. Sun, K-means hashing: an affinity-preserving quantization method for learning binary compact codes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2938–2945.
[11] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for scalable image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3424–3431.
[12] J. Song, Y. Yang, Y. Yang, Z. Huang, H.T. Shen, Inter-media hashing for large-scale retrieval from heterogeneous data sources, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, 2013, pp. 785–796.
[13] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2075–2082.
[14] J. Zhou, G. Ding, Y. Guo, Latent semantic sparse hashing for cross-modal similarity search, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2014, pp. 415–424.
[15] D. Zhang, F. Wang, L. Si, Composite hashing with multiple information sources, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 225–234.
[16] M.M. Bronstein, A.M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing (2010).
[17] K. Wang, R. He, W. Wang, L. Wang, T. Tan, Learning coupled feature spaces for cross-modal matching, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2088–2095.
[18] Z. Lin, G. Ding, M. Hu, J. Wang, Semantics-preserving hashing for cross-view retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3864–3872.
[19] J. Zhou, G. Ding, Y. Guo, Q. Liu, X. Dong, Kernel-based supervised hashing for cross-view similarity search, in: Proceedings of the IEEE International Conference on Multimedia and Expo, IEEE, 2014, pp. 1–6.
[20] X. Xu, L. He, A. Shimada, R.-i. Taniguchi, H. Lu, Learning unified binary codes for cross-modal retrieval via latent semantic hashing, Neurocomputing (2016).
[21] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 689–696.
[22] N. Srivastava, R. Salakhutdinov, Learning representations for multimodal data with deep belief nets, in: Proceedings of International Conference on Machine Learning Workshop, 2012.
[23] W. Wang, B.C. Ooi, X. Yang, D. Zhang, Y. Zhuang, Effective multi-modal retrieval based on stacked auto-encoders, Proc. VLDB Endow. 7 (8) (2014) 649–660.
[24] Y. Kang, S. Kim, S. Choi, Deep learning to hash with multiple representations, in: Proceedings of the 12th International Conference on Data Mining, IEEE, 2012, pp. 930–935.
[25] D. Wang, P. Cui, M. Ou, W. Zhu, Deep multimodal hashing with orthogonal regularization, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2015.
[26] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing, IEEE Trans. Pattern Anal. Mach. Intell. 34 (6) (2012) 1092–1104.
[27] J.C. Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G.R. Lanckriet, R. Levy, N. Vasconcelos, On the role of correlation and abstraction in cross-modal multimedia retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 521–535.

[28] A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin, Context-based vision system for place and object recognition, in: Proceedings Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 273–280.
[29] 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: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 48.
[30] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Heng-Tao Shen, Deep Region Hashing for Efficient Large-scale Instance Search from Images, arXiv preprint arXiv:1701.07901
[31] Zhihua Xia, Xinhui Wang, Liangao Zhang, Zhan Qin, Xingming Sun, Kui Ren, A Privacy-preserving and Copy-deterrence Content-based Image Retrieval Scheme in Cloud Computing, IEEE Transactions on Information Forensics and Security 11 (11) (2016) 2594–2608.
[32] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, Heng Tao Shen, A Survey on Learning to Hashing (2016) arXiv preprint arXiv:1606.00185.

Xiangpeng Li is an undergraduate student in the school of Computer Science and Engineering, University of Electronic Science and Technology. His research interest is image and video retrieval.

Lianli Gao's background lines in Semantic Web, Machine Learning and Computer Vision. She received my PhD degree in Information Technology from University of Queensland, Brisbane, Australia. Currently, am an Associate Professor working at the University of Electronic Science and Technology of China.

Xing Xu is a member of the Faculty of Information Science and Electrical Engineering, Kyushu University. He works as a research technician in Department of Advance Information Technology. He received a PhD degree from Kyushu University, Japan in 2015. His research is in the fields of pattern recognition and multimedia processing, with particular interests in large scale image/video annotation and retrieval, along with machine learning techniques. He is also interested in the related mobile applications of multimedia processing.

Jie Shao his PhD degree in Information Technology is obtained from The University of Queensland, Australia. Currently he is professor at the University of Electronic Science and Technology of China. His research interest includes trajectory data analysis and computer vision.

Fumin Shen received his B.S. and Ph.D. degree from Shandong University and Nanjing University of Science and Technology, China, in 2007 and 2014, respectively. Currently he is a lecturer in school of Computer Science and Engineering, University of Electronic of Science and Technology of China, China. His major research interests include computer vision and machine learning, including face recognition, image analysis, hashing methods, and robust statistics with its applications in computer vision.

Jingkuan Song received his PhD degree in Information Technology from The University of Queensland, Australia. He received his BS degree in Software Engineering from University of Electronic Science and Technology of China. Currently, he is a postdoctoral researcher in the Dept. of Information Engineering and Computer Science, University of Trento, Italy. His research interest includes large-scale multimedia search and machine learning.