Pattern Recognition 107 (2020) 107479

Contents lists available at ScienceDirect

Pattern Recognition

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

Joint and individual matrix factorization hashing for large-scale cross-modal retrieval

Di Wang[a], Quan Wang[a], Lihuo He[b], Xinbo Gao[b,c,*], Yumin Tian[a]

[a] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[b] Video and Image Processing System (VIPs) Lab, School of Electronic Engineering, Xidian University, Xi'an 710071, China
[c] The Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

ARTICLE INFO

Article history:
Received 3 November 2019
Revised 13 April 2020
Accepted 28 May 2020
Available online 31 May 2020

Keywords:
Hashing
Multimodal
Retrieval
Cross-modal
Matrix factorization

ABSTRACT

Multimodal hashing methods have gained considerable attention in recent years due to their effectiveness and efficiency for cross-modal similarity searches. Existing multimodal hashing methods either learn unified hash codes for different modalities or learn individual hash codes for each modality and then explore cross-correlations between them. Generally, learning unified hash codes tends to preserve the shared properties of multimodal data and learning individual hash codes tends to preserve the specific properties of each modality. There remains a crucial bottleneck regarding how to learn hash codes that simultaneously preserve the shared properties and specific properties of multimodal data. Therefore, we present a joint and individual matrix factorization hashing (JIMFH) method, which not only learns unified hash codes for multimodal data to preserve their common properties but also learns individual hash codes for each modality to retain its specific properties. The proposed JIMFH learns unified hash codes by joint matrix factorization, which jointly factorizes all modalities into a shared latent semantic space. In addition, JIMFH learns individual hash codes by individual matrix factorization, which separately factorizes each modality into a modal-specific latent semantic space. Finally, unified hash codes and individual hash codes are combined to obtain the final hash codes. In this way, hash codes learned by JIMFH can preserve both the shared properties and specific properties of multimodal data, and therefore the retrieval performance is enhanced. Comprehensive experiments show that the proposed JIMFH performs much better than many state-of-the-art methods on cross-modal retrieval applications.

1. Introduction

During recent years, with the unprecedented growth of multimedia data on the Internet, efficient cross-modal similarity search has become an interesting but challenging topic, that has attracted much attention [1,2]. Due to the benefits of low storage cost and fast query speed, multimodal hashing methods have been gaining more popularity than traditional real-valued cross-modal similarity search methods [3–5]. The key concept of multimodal hashing is transforming high-dimensional multimodal data (e.g., images, documents, and videos) into a set of equal-length compact binary codes that preserve the cross-modality similarities of the original data [6]. In this way, the cross-modality similarity distances between original multimodal data can be directly obtained by computing Hamming distances between binary codes [7].

Multimodal hashing methods can be grouped into two categories: unsupervised methods and supervised methods. Supervised multimodal hashing methods often use available pairwise constraints or semantic labels for learning hash functions. Many traditional supervised multimodal hashing methods attempt to encode multimodal data as compact binary codes whose Hamming distances approximate the pairwise similarities that are derived from labels or constraints [8–10]. To enhance the discriminability of hash codes, many label-based supervised multimodal hashing methods have been proposed [11–13]. Such methods directly utilize labels to learn discriminative binary codes. Above traditional methods learn hash codes through a two-step process. They first extract hand-crafted features from multimodal data and then learn hash functions to transform features into binary codes. Deep supervised multimodal methods usually integrate deep feature learning and hash code learning into an end-to-end deep neural network and can achieve good performance [14–16].

While supervised multimodal hashing methods can achieve significant performance, supervised information is often challenging

* Corresponding author.
E-mail address: xbgao@mail.xidian.edu.cn (X. Gao).

https://doi.org/10.1016/j.patcog.2020.107479
0031-3203/

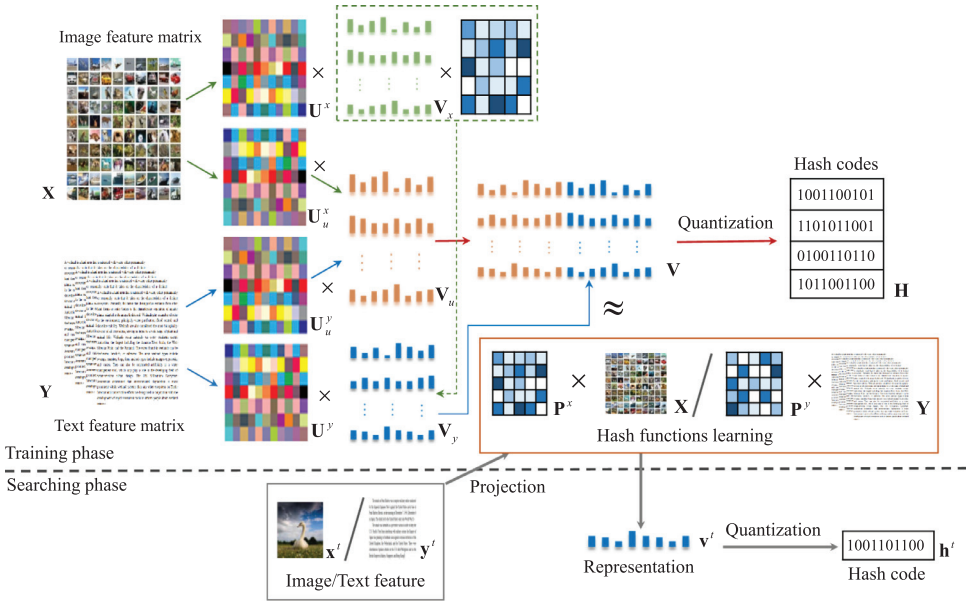

Fig. 1. Framework of joint and individual matrix factorization hashing. First, it transforms multimodal data into an unified latent semantic space jointly to obtain unified representations, and also transforms multimodal data into latent semantic spaces individually to obtain individual representations. Then, the learned unified and individual representations are combined and quantized to obtain the final hash codes. Finally, It learns linear projection matrices for out-of-sample extensions.

and time consuming to obtain, as labeling samples requires significant efforts of human annotators. Unsupervised multimodal hashing methods usually learn binary codes from data distributions without referring to supervised information. Graph-based multimodal hashing methods learn hash codes by constructing similarity graphs[17-[20]]. However, these methods suffer from high training complexity for computing the similarity graphs. Matrix factorization-based methods overcome these drawbacks by transforming data with different modalities into a common latent semantic space through matrix factorization [21–24]. Very recently, unsupervised deep multimodal hashing inspired by the successful applications of deep learning has gained wide attention [25–27]. Such methods usually have good performance. However, the training complexities of these deep methods are very high.

Although the significant progress that has been made by existing unsupervised multimodal hashing works, many challenges remain to be addressed. Many unsupervised multimodal hashing methods map multimodal data into a shared latent space to learn unified hash codes; that is, each data pair shares an identical binary code [21,24]. These methods usually preserve shared properties of multimodal data and cannot preserve the specific properties of each modality. Different from them, some other methods map each modality into an individual latent space to learn individual hash codes; that is, different modalities in a data pair have different binary codes, and then explore cross-correlations between them to establish relations [17–19,22,23]. In such a way, the learned individual hash codes tend to preserve specific properties of each modality and cannot preserve the shared properties. Therefore, how to learn hash codes that simultaneously preserve shared properties and specific properties of multimodal data remains to be studied further.

To address the above mentioned challenges, a novel method termed joint and individual matrix factorization hashing (JIMFH) is proposed for large-scale cross-modal retrieval, which aims to preserve the shared and specific properties of multimodal data for hash codes simultaneously. It utilizes joint matrix factorization and individual matrix factorization for processing multimodal data, in which it not only learns unified hash codes for different modalities in a unified latent subspace to preserve their shared properties but also learns individual hash codes for each modality to preserve its specific property. Fig. 1 illustrates the flowchart of JIMFH. It transforms multimodal data into an unified latent semantic space jointly to obtain unified representations, and transforms multimodal data into latent semantic spaces individually to obtain individual representations. Then, the learned unified and individual representations are combined to form the synthesized representations. Next, by quantizing the synthesized representations, hash codes can be obtained. In addition, it learns linear projection matrices for out-of-sample extensions. Consequently, hash codes of queries can be generated by linear projections in the searching phase. The main contributions of JIMFH are summarized as follows:

- A joint and individual matrix factorization hashing method is proposed to simultaneously learn unified and individual hash codes for multimodal data. Hence, the shared and specific properties of multimodal data are both preserved for the learned hash codes and the cross-modal retrieval performance is improved.
- An effective optimization algorithm is proposed to solve the proposed method. In addition, its theoretical analysis is provided in detail.
- Extensive experimental results on three multimodal data sets highlight the superiority of the proposed method over state-of-the-art unsupervised multimodal hashing methods.

The rest of this paper is organized as follows. Section 2 reviews some related works. Section 3 introduces the proposed joint and individual matrix factorization hashing method with its theoretical analysis. Section 4 introduces the experimental results and analysis. Finally, we present the conclusions in Section 5.

2. Related work

In recent years, with the rapid development of multimedia data (such as images, texts, and videos) on the Internet, cross-modal retrieval methods have received considerable attention. These methods enable users to retrieve relevant data across different media types. For example, one can use an image to retrieve relevant texts

or videos. As multimedia data have different modalities, directly measuring similarities between different modalities are infeasible. Therefore, measuring the similarities between different modalities of data is the core issue for cross-modal retrieval methods. To address this issue, various methods have been proposed. Generally, these methods can be classified into two main groups: 1) real-valued cross-modal retrieval methods, and 2) binary cross-modal retrieval methods (i.e., multimodal hashing methods). In this section, we review a number of representative methods of these two kinds of cross-modal retrieval methods.

2.1. Real-valued cross-modal retrieval methods

Real-valued cross-modal retrieval methods learn a real-valued common representation space for multimodal data in which data with different modalities can be measured directly. According to how the common representation space is learned, the methods can be categorized into subspace learning methods, topic model methods, and deep learning methods. The representative subspace learning method is the canonical correlation analysis (CCA)-based cross-modal retrieval method [28]. It learns a common subspace by maximizing the correlation between different modalities with CCA, such that the similarities between different modalities can be measured in the learned common subspace. In addition to the CCA-based method, many other methods based on various subspace learning methods have been proposed. These methods include dictionary learning-based methods [29,30], and multiview analysis-based methods [31]. Topic model methods discover latent topics of multimodal data to enable cross-modal retrieval [32,33]. Deep learning methods usually design deep neural networks to transform multimodal data into unified representations and then the similarities between different modalities can be obtained by calculating distances between deep representations [34,35].

2.2. Multimodal hashing methods

Generally, real-valued cross-modal retrieval methods are time-consuming for large-scale data sets. To speed up cross-modal retrieval, binary cross-modal retrieval methods; namely, multimodal hashing methods, have been proposed recently. These methods transform high-dimensional multimodal data into a set of equal-length compact binary codes that preserve the cross-modality similarities of original data. As similarity measurements between these binary codes only involve efficient bit-count operations, they can thus be computed very efficiently. According to how much supervised information is used, multimodal hashing methods can be grouped into three categories: supervised methods, semi-supervised methods, and unsupervised methods.

Supervised multimodal hashing methods learn hash codes that preserve semantic similarities obtained from the semantic labels or pairwise constraints of training points. Many supervised multimodal hashing methods let the to -be-learned hash codes approximate the pairwise similarity matrices that are constructed by available supervised information, such as semantic correlation maximization (SCM) [8], and semantics-preserving hashing (SEPH) [9]. These methods preserve pariwise similarities for hash codes and neglect the discriminative information reflected by labels. To address this problem, some other methods are proposed to directly utilize labels to learn discriminative hash codes, such as multimodal discriminative binary embedding (MDBE) [11], label consistent matrix factorization hashing (LCMFH) [13], and discrete cross-modal hashing (DCH) [12]. Recently, deep supervised multimodal hashing methods have achieved favorable performance by taking advantage of deep neural networks, which can capture nonlinear correlations among multimodal data, such as deep cross-modal hashing (DCMH) [14], and self-supervised adversarial hashing (SSAH) [15].

Supervised methods usually require supervised information of the entire training set, which is usually challenging and time consuming to obtain. Semi-supervised multimodal hashing methods utilize partially labeled training data and the remaining unlabled data to learn the hash model. Weakly supervised multimodal hashing (WSMH) explores the semantic structure, the local discriminative structure and the geometric structure to learn hash functions [36]. Semi-supervised cross-modal hashing by generative adversarial network (SCH-GAN) uses a generative adversarial network to fit the relevance distribution of unlabeled data, and tries to select margin examples from unlabeled data of one modality, giving a query of another modality [37].

Unlike supervised and semi-supervised methods, unsupervised multimodal hashing methods usually learn hash codes from data distributions without referring to supervised information. Cross-view hashing (CVH) [17] learns hash codes by maximizing the intra-modality and inter-modality similarities between data pairs. Inter-media hashing (IMH) [18] constructs inter-media and intra-media graphs to learn hash codes by preserving graph similarities. Linear cross-modal hashing (LCMH) [19] avoids the large-graph construction process in IMH by representing data points with a smaller approximation graph. The above methods are all graph-based methods. However, these methods suffer from high training complexity for computing the similarity graphs. To avoid this problem, matrix factorization-based methods are proposed, such as collective matrix factorization hashing (CMFH) [21], latent semantic sparse hashing (LSSH) [23], semantic topic multimodal hashing (STMH) [22], and robust and flexible discrete hashing (RFDH) [24]. CMFH, LSSH, STMH and RFDH learn hash codes by collective matrix factorization, sparse representation, robust matrix factorization and topic model, and robust discrete matrix factorization, respectively. Among these four methods, CMFH, STMH and RFDH learn unified hash codes that tend to preserve the shared properties of multimodal data. LSSH learns individual hash codes that tend to preserve the specific properties of each modality. Therefore, how to learn hash codes that simultaneously preserve the shared properties and specific properties of multimodal data remains to be solved. In this paper, a new unsupervised multimodal hashing method named joint and individual matrix factorization hashing (JIMFH) is proposed to simultaneously preserve the shared properties and specific properties of multimodal data. There are also some works proposed to use deep neural networks to learn hash codes in an unsupervised way [25–27]. Such methods usually have good performance and high training complexities. In this paper, we mainly focus on shallow unsupervised multimodal hashing methods.

3. Joint and individual matrix factorization hashing

In the following, we first detail the motivation and formulation of joint and individual matrix factorization hashing (JIMFH), and then present its optimization method and theoretical analysis. For a clear presentation, we first introduce the proposed JIMFH in bimodal data (i.e., image and text). It can be easily extended to multimodal cases without loss of generality.

3.1. Notations and problem formulation

Let $\mathcal{O} = \{o_i\}_{i=1}^{n}$, $o_i = (\mathbf{x}_i, \mathbf{y}_i)$ denote a training bimodal data set with n instances, where $\mathbf{x}_i \in \mathbb{R}^{d_x}$ and $\mathbf{y}_i \in \mathbb{R}^{d_y}$ are image and text feature vectors respectively for the ith instance o_i. d_x and d_y are the dimensionalities of the image and text spaces, respectively, usually, $d_x \neq d_y$. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d_x \times n}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \mathbb{R}^{d_y \times n}$ denote the image and text matrices formed by image and

text feature vectors in $\mathcal{O}$, respectively. Without loss of generality, we assume that image and text features have zero mean, i.e., $\sum_{i=1}^{n}\mathbf{x}_i = \mathbf{0}$ and $\sum_{i=1}^{n}\mathbf{y}_i = \mathbf{0}$.

Given code length k, JIMFH aims to learn binary codes $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n] \in \mathbb{R}^{k\times n}$, where $\mathbf{b}_i \in \{0, 1\}^k$, for training bimodal data set $\mathcal{O}$, which preserves the similarities of original feature vectors with high probability.

3.2. Framework overview

In general, the main challenge of cross-modal retrieval is the 'modality gap', which means multimodal data may have completely different feature representations, dimensionalities, distributions, and statistical properties, resulting in direct similarity measurements between different modalities being infeasible. However, fortunately, multimodal data that describe the same instance usually have close semantic meanings. Therefore, we can transform multimodal data into a shared latent semantic space to bridge the gap between different modalities. For multimodal data (e.g., images and texts), due to their heterogeneity, they usually have their own specific properties, e.g., text features are usually sparser than image features. Therefore, it is better to preserve the different specific properties of data with different modalities in the transformed latent semantic space. In addition, it is reasonable to assume that image-text pairs that describe the same instances have some shared properties in latent semantic space. Accordingly, we expect the representations of multimodal data in the transformed latent semantic space to share the common properties and preserve the specific properties of each modality simultaneously. In this paper, we use matrix factorization [38–40] to learn semantic spaces hidden in original multimodal data, as its effectiveness in cross-modal retrieval has been verified by many multimodal hashing methods [21–24].

The framework of the proposed JIMFH is illustrated in Fig. 1. It consists of a training phase and a searching phase. In the training phase, it simultaneously uses joint matrix factorization and individual matrix factorization to explore latent semantic spaces to connect different modalities. Then, the representations of multimodal data in latent semantic spaces are combined and quantified to obtain training hash codes. In addition, linear projection matrices are learned for transforming queries into binary codes. In the searching phase, given a query, we can use the corresponding linear projection matrix to generate its hash code. In the following subsections, we describe each part of the JIMFH in detail.

3.3. Joint matrix factorization

Collective matrix factorization (CMF) is a simple and effective algorithm for learning cross-modal relations among multimodal data [38]. Based on CMF, Ding *et al.* proposed the collective matrix factorization hashing (CMFH) [21], which is the pioneering work of matrix factorization-based multimodal hashing method and achieves promising performance. In the proposed JIMFH, we also use CMF to jointly factorize multimodal data into an unified latent semantic space.

Given feature matrices $\mathbf{X}$ and $\mathbf{Y}$, the joint matrix factorization (JMF) part of JIMFH aims to find three matrices $\mathbf{U}_u^x = [\mathbf{u}_{u1}^x, \mathbf{u}_{u2}^x, \cdots, \mathbf{u}_{uk_u}^x] \in \mathbb{R}^{d_x\times k_u}$, $\mathbf{U}_u^y = [\mathbf{u}_{u1}^y, \mathbf{u}_{u2}^y, \cdots, \mathbf{u}_{uk_u}^y] \in \mathbb{R}^{d_y\times k_u}$ and $\mathbf{V}_u = [\mathbf{v}_{u1}, \mathbf{v}_{u2}, \cdots, \mathbf{v}_{un}] \in \mathbb{R}^{k_u\times n}$, whose product approximates $\mathbf{X}$ and $\mathbf{Y}$ as closely as possible

$$\begin{cases}\mathbf{X} \approx \mathbf{U}_u^x\mathbf{V}_u \\ \mathbf{Y} \approx \mathbf{U}_u^y\mathbf{V}_u\end{cases}, \tag{1}$$

where k_u is the dimensionality of the unified latent semantic space and is an integer between 1 and $k-1$.

To measure the quality of the approximate decomposition defined in (1), we adopt the widely used Euclidean distance metric. Then, the objective function of the JMF in the JIMFH is defined as

$$\min_{\mathbf{U}_u^x,\mathbf{U}_u^y,\mathbf{V}_u} \lambda\|\mathbf{X} - \mathbf{U}_u^x\mathbf{V}_u\|_F^2 + (1-\lambda)\left\|\mathbf{Y} - \mathbf{U}_u^y\mathbf{V}_u\right\|_F^2, \tag{2}$$

where $\mathbf{U}_u^x$ and $\mathbf{U}_u^y$ are latent vector matrices for image and text, respectively, $\mathbf{V}_u$ is the unified representation matrix, and $\lambda \in (0, 1)$ is a parameter that controls the weights of image and text.

Through joint matrix factorization, image and text features are approximated by linear combinations of latent vectors $\mathbf{U}_u^x$ and $\mathbf{U}_u^y$, weighted by unified representations $\mathbf{V}_u$. Actually, $\mathbf{U}_u^x$ and $\mathbf{U}_u^y$ can be seen as basis vectors that form the latent semantic spaces of images and texts. Then, $\mathbf{V}_u$ can be regarded as unified representations of image and text in latent semantic space. As the latent semantic representation of one image-text pair is forced to be identical, strong connections between image and text are built. Therefore, the shared properties of multimodal data are preserved for hash codes, which are quantized by $\mathbf{V}_u$.

Note that the existing CMFH [21], RFDH [24], and LCMFH [13] methods also use CMF to learn unified latent semantic space for different modalities. However, they mainly use CMF to learn entire hash codes. Whereas, the proposed JIMFH only uses CMF to learn a portion of the final hash codes that retain the shared properties of multimodal data. In addition, LCMFH adds a label consistent constraint on CMF to ensure multimodal data from the same category share the same representation. RFDH aims to learn more robust hash codes by using the $\ell_{2,1}$-norm in the objective function. Therefore, the motivation for the proposed JIMFH method is different from these methods.

3.4. Individual matrix factorization

The proposed JIMFH utilizes JMF to learn unified hash codes that preserve shared properties of multimodal data. To preserve the specific properties of each modality, JIMFH simultaneously factorizes each modality respectively. Analogous to the JMF, the objective function of the individual matrix factorization (IMF) is as follows:

$$\min_{\mathbf{U}^x,\mathbf{U}^y,\mathbf{V}_x,\mathbf{V}_y,\mathbf{R}} \lambda\|\mathbf{X} - \mathbf{U}^x\mathbf{V}_x\|_F^2 + (1-\lambda)\|\mathbf{Y} - \mathbf{U}^y\mathbf{V}_y\|_F^2 + \mu\|\mathbf{V}_y - \mathbf{R}\mathbf{V}_x\|_F^2, \tag{3}$$

where $\mathbf{U}^x \in \mathbb{R}^{d_x\times k_s}$ and $\mathbf{V}_x \in \mathbb{R}^{k_s\times n}$ are the latent vector matrix and representation matrix for images, $\mathbf{U}^y \in \mathbb{R}^{d_y\times k_s}$ and $\mathbf{V}_y \in \mathbb{R}^{k_s\times n}$ are the latent vector matrix and representation matrix for texts, k_s is the dimensionality of the specific latent semantic spaces and $k_s = k - k_u$, $\mathbf{R} \in \mathbb{R}^{k_s\times k_s}$ is the correlation matrix between images and texts, and μ is a nonnegative parameter controlling the contribution of the correlation matching term $\|\mathbf{V}_y - \mathbf{R}\mathbf{V}_x\|_F^2$.

In (3), the image matrix and text matrix are decomposed individually. Therefore, representation matrices $\mathbf{V}_x$ and $\mathbf{V}_y$ preserve the specific properties of the image and text, respectively. Through the correlation matrix $\mathbf{R}$, the specific representations of image and text are connected.

3.5. Overall objective function

The core concept of JIMFH is to preserve the shared and specific properties of different modalities through learning unified and individual hash codes simultaneously. To achieve this, it factorizes different modalities jointly and individually through joint matrix factorization and individual matrix factorization. Therefore, combining the joint matrix factorization term given in (2), the individual matrix factorization term given in (3) and the regularization

term, the overall objective function of JIMFH is formulated as

$$\min \mathcal{F}\left(\mathbf{U}_u^x, \mathbf{U}_u^y, \mathbf{V}_u, \mathbf{U}^x, \mathbf{U}^y, \mathbf{V}_x, \mathbf{V}_y, \mathbf{R}\right), \tag{4}$$

where

$$\begin{aligned}\mathcal{F} = &\lambda\left(\|\mathbf{X} - \mathbf{U}_u^x\mathbf{V}_u\|_F^2 + \|\mathbf{X} - \mathbf{U}^x\mathbf{V}_x\|_F^2\right) + \mu\|\mathbf{V}_y - \mathbf{R}\mathbf{V}_x\|_F^2 \\ &+(1-\lambda)\left(\left\|\mathbf{Y} - \mathbf{U}_u^y\mathbf{V}_u\right\|_F^2 + \|\mathbf{Y} - \mathbf{U}^y\mathbf{V}_y\|_F^2\right) \\ &+\gamma R\left(\mathbf{U}_u^x, \mathbf{U}_u^y, \mathbf{V}_u, \mathbf{U}^x, \mathbf{U}^y, \mathbf{V}_x, \mathbf{V}_y, \mathbf{R}\right),\end{aligned} \tag{5}$$

where $R(\cdot) = \|\cdot\|_F^2$ is the regularization term to avoid overfitting, and γ is a parameter controlling the weight of the regularization term.

After optimizing (4), we can obtain unified representations $\mathbf{V}_u$ and specific representations $\mathbf{V}_x$ and $\mathbf{V}_y$. As representations $\mathbf{V}_x$ can be transformed into $\mathbf{V}_y$ by correlation matrix $\mathbf{R}$, we can use $\mathbf{V}_y$ to represent the specific representations of image-text pairs. Then, the latent semantic representations $\mathbf{V}$ of training data $\mathcal{O}$ can be obtained by combining unified representations $\mathbf{V}_u$ and specific representations $\mathbf{V}_y$ as

$$\mathbf{V} = \begin{bmatrix}\mathbf{V}_u \\ \mathbf{V}_y\end{bmatrix}. \tag{6}$$

Then, we can obtain hash codes $\mathbf{B}$ by quantifying latent semantic representations $\mathbf{V}$ as

$$\mathbf{B} = \mathrm{sgn}(\mathbf{V}), \tag{7}$$

where $\mathrm{sgn}(\cdot)$ is an element-wise sign function. Therefore, the learned hash codes preserve the shared and specific properties of multimodal data simultaneously.

3.6. Optimization

The optimization problem of (4) is non-convex for all the matrix variables together, but is convex with respect to any one of eight matrix variables when others are fixed. In this subsection, an iterative algorithm is introduced to solve (4) by iteratively updating the following steps.

1. **Updating** $\mathbf{U}_u^x$. Let the partial derivative of $\mathcal{F}$ with respect to $\mathbf{U}_u^x$ equal zero, we can obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}_u^x} = 2\lambda\mathbf{U}_u^x\mathbf{V}_u\mathbf{V}_u^T + 2\gamma\mathbf{U}_u^x - 2\lambda\mathbf{X}\mathbf{V}_u^T = 0. \tag{8}$$

Then, the closed-form solution of $\mathbf{U}_u^x$ is obtained.

$$\mathbf{U}_u^x = \mathbf{X}\mathbf{V}_u^T\left(\mathbf{V}_u\mathbf{V}_u^T + \frac{\gamma}{\lambda}\mathbf{I}\right)^{-1}, \tag{9}$$

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

2. **Updating** $\mathbf{U}_u^y$. Analogous to $\mathbf{U}_u^x$, we can obtain the closed-form solution of $\mathbf{U}_u^y$.

$$\mathbf{U}_u^y = \mathbf{Y}\mathbf{V}_u^T\left(\mathbf{V}_u\mathbf{V}_u^T + \frac{\gamma}{1-\lambda}\mathbf{I}\right)^{-1}. \tag{10}$$

3. **Updating** $\mathbf{V}_u$. Let the partial derivative of $\mathcal{F}$ with respect to $\mathbf{V}_u$ equal zero, we can obtain

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{V}_u} = &2\lambda\mathbf{U}_u^{xT}\mathbf{U}_u^x\mathbf{V}_u + 2(1-\lambda)\mathbf{U}_u^{yT}\mathbf{U}_u^y\mathbf{V}_u + 2\gamma\mathbf{V}_u \\ &-2\lambda\mathbf{U}_u^{xT}\mathbf{X} - 2\lambda\mathbf{U}_u^{yT}\mathbf{Y} = 0.\end{aligned} \tag{11}$$

Then, the closed-form solution of $\mathbf{V}_u$ is obtained.

$$\mathbf{V}_u = \left(\lambda\mathbf{U}_u^{xT}\mathbf{U}_u^x + (1-\lambda)\mathbf{U}_u^{yT}\mathbf{U}_u^y + \gamma\mathbf{I}\right)^{-1}\left(\lambda\mathbf{U}_u^{xT}\mathbf{X} + (1-\lambda)\mathbf{U}_u^{yT}\mathbf{Y}\right). \tag{12}$$

4. **Updating** $\mathbf{U}^x$. Analogous to $\mathbf{U}_u^x$, the closed-form solution of $\mathbf{U}^x$ is obtained.

$$\mathbf{U}^x = \mathbf{X}\mathbf{V}_x^T\left(\mathbf{V}_x\mathbf{V}_x^T + \frac{\gamma}{\lambda}\mathbf{I}\right)^{-1}. \tag{13}$$

5. **Updating** $\mathbf{U}^y$. Analogous to $\mathbf{U}_u^x$, the closed-form solution of $\mathbf{U}^y$ is obtained.

$$\mathbf{U}^y = \mathbf{Y}\mathbf{V}_y^T\left(\mathbf{V}_y\mathbf{V}_y^T + \frac{\gamma}{(1-\lambda)}\mathbf{I}\right)^{-1}. \tag{14}$$

6. **Updating** $\mathbf{V}_x$. Analogous to $\mathbf{V}_u$, the closed-form solution of $\mathbf{V}_x$ is obtained.

$$\mathbf{V}_x = \left(\lambda\mathbf{U}^{xT}\mathbf{U}^x + \mu\mathbf{R}^T\mathbf{R} + \gamma\mathbf{I}\right)^{-1}\left(\lambda\mathbf{U}^{xT}\mathbf{X} + \mu\mathbf{R}^T\mathbf{V}_y\right). \tag{15}$$

7. **Updating** $\mathbf{V}_y$. Analogous to $\mathbf{V}_u$, the closed-form solution of $\mathbf{V}_y$ is obtained.

$$\mathbf{V}_y = \left((1-\lambda)\mathbf{U}^{yT}\mathbf{U}^y + (\mu+\gamma)\mathbf{I}\right)^{-1}\left((1-\lambda)\mathbf{U}^{yT}\mathbf{Y} + \mu\mathbf{R}\mathbf{V}_x\right). \tag{16}$$

8. **Updating** $\mathbf{R}$. Let the partial derivative of $\mathcal{F}$ with respect to $\mathbf{R}$ equal zero, we can obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{R}} = 2\mu\mathbf{R}\mathbf{V}_x\mathbf{V}_x^T + 2\gamma\mathbf{R} - 2\mu\mathbf{V}_y^T\mathbf{V}_x = 0. \tag{17}$$

Then, the closed-form solution of $\mathbf{R}$ is obtained.

$$\mathbf{R} = \mathbf{V}_y\mathbf{V}_x^T\left(\mathbf{V}_x\mathbf{V}_x^T + \frac{\gamma}{\mu}\mathbf{I}\right)^{-1}. \tag{18}$$

The final solution of $\mathcal{F}$ is obtained by iteratively updating eigth matrix variables until the objective function converges or reaches the preset maximum number of iterations. The whole training process of JIMFH introduced above is summarized in Algorithm 1.

3.7. Convergence analysis

In this subsection, the convergence of the proposed iterative method is analyzed.

In each iteration t, according to Step 3.1 in Algorithm 1, we have

$$\mathbf{U}_u^{x(t+1)} = \min_{\mathbf{U}_u^x} \mathcal{F}\left(\mathbf{U}_u^x, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right). \tag{19}$$

Then, we obtain the following inequality

$$\begin{aligned}&\mathcal{F}\left(\mathbf{U}_u^{x(t+1)}, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right) \\ &\le \mathcal{F}\left(\mathbf{U}_u^{x(t)}, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right).\end{aligned} \tag{20}$$

Similarly, according to Step 3.2, we have

$$\mathbf{U}_u^{y(t+1)} = \min_{\mathbf{U}_u^y} \mathcal{F}\left(\mathbf{U}_u^{x(t+1)}, \mathbf{U}_u^y, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right). \tag{21}$$

Then, we obtain the following inequality

$$\begin{aligned}&\mathcal{F}\left(\mathbf{U}_u^{x(t+1)}, \mathbf{U}_u^{y(t+1)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right) \\ &\le \mathcal{F}\left(\mathbf{U}_u^{x(t+1)}, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right) \\ &\le \mathcal{F}\left(\mathbf{U}_u^{x(t)}, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right).\end{aligned} \tag{22}$$

Similarly, the objective function $\mathcal{F}$ is also monotonically reduced from step 3.3 to step 3.8 in Algorithm 1. Therefore, we have

$$\begin{aligned}&\mathcal{F}\left(\mathbf{U}_u^{x(t+1)}, \mathbf{U}_u^{y(t+1)}, \mathbf{V}_u{}^{(t+1)}, \mathbf{U}^{x(t+1)}, \mathbf{U}^{y(t+1)}, \mathbf{V}_x{}^{(t+1)}, \mathbf{V}_y{}^{(t+1)}, \mathbf{R}^{(t+1)}\right) \\ &\le \mathcal{F}\left(\mathbf{U}_u^{x(t)}, \mathbf{U}_u^{y(t)}, \mathbf{V}_u{}^{(t)}, \mathbf{U}^{x(t)}, \mathbf{U}^{y(t)}, \mathbf{V}_x{}^{(t)}, \mathbf{V}_y{}^{(t)}, \mathbf{R}^{(t)}\right).\end{aligned}$$

Algorithm 1 Joint and individual matrix factorization hashing.

Training stage

Input: image and text feature matrices $\mathbf{X}$ and $\mathbf{Y}$, hash code length k, unified hash code lenght k_u, parameters λ, μ and γ.

Output: hash codes $\mathbf{B}$, image and text projection matrices $\mathbf{P}^x$ and $\mathbf{P}^y$.

Procedure:

1. Centralize $\mathbf{X}$ and $\mathbf{Y}$ by means.
2. Initialize $\mathbf{V}_u$, $\mathbf{V}_x$, $\mathbf{V}_y$ and $\mathbf{R}$ by random matrices.

Repeat

3.1 Update $\mathbf{U}_u^x$ by Eq. (9).
3.2 Update $\mathbf{U}_u^y$ by Eq. (10).
3.3 Update $\mathbf{V}_u$ by Eq. (12).
3.4 Update $\mathbf{U}^x$ by Eq. (13).
3.5 Update $\mathbf{U}^y$ by Eq. (14).
3.6 Update $\mathbf{V}_x$ by Eq. (15).
3.7 Update $\mathbf{V}_y$ by Eq. (16).
3.8 Update $\mathbf{R}$ by Eq. (18).

Until Convergence.

4. Calculate latent semantic representations by $\mathbf{V} = [\mathbf{V}_u{}^T, \mathbf{V}_y{}^T]^T$.
5. Calculate hash codes by $\mathbf{B} = \mathrm{sgn}(\mathbf{V})$.
6. Learn projection matrices $\mathbf{P}^x$ and $\mathbf{P}^y$ by Eq. (32) and Eq. (33).

Querying stage

Input: feature vector $\mathbf{x}^t$ or $\mathbf{y}^t$, projection matrices $\mathbf{P}^x$ and $\mathbf{P}^y$.

Output: hash code $\mathbf{b}^t$.

Procedure:

1. Centralize $\mathbf{x}^t$ or $\mathbf{y}^t$ by means.
2. For $\mathbf{x}^t$: calculate hash code by $\mathbf{b}^t = \mathrm{sgn}(\mathbf{P}^x\mathbf{x}^t)$.
For $\mathbf{y}^t$: calculate hash code by $\mathbf{b}^t = \mathrm{sgn}(\mathbf{P}^y\mathbf{y}^t)$.

(23)

That is

$$\mathcal{F}^{(t+1)} \leq \mathcal{F}^{(t)}. \tag{24}$$

According to the above analysis, the objective value of $\mathcal{F}$ is decreasing under each iteration and will converge to a stable point.

3.8. Hash functions learning

Hash codes for training data $\mathcal{O}$ can be directly obtained by the above method. However, how to generate hash codes for arbitrary samples outside the training set still remains to be solved. For query samples, modal-specific hash functions $f(\mathbf{x})$ and $f(\mathbf{y})$ are learned to generate hash codes for query images and texts respectively. The commonly used linear projection hashing function form is adopted in this paper, which is defined as

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{P}^x\mathbf{x}), \tag{25}$$

$$f(\mathbf{y}) = \mathrm{sgn}(\mathbf{P}^y\mathbf{y}), \tag{26}$$

where $\mathbf{P}^x$ and $\mathbf{P}^y$ are linear projection matrices.

Given training image and text matrices $\mathbf{X}$ and $\mathbf{Y}$ and their hash codes $\mathbf{B}$, hash functions should transform training data into their hash code, that is,

$$\mathbf{B} = f(\mathbf{X}) = \mathrm{sgn}(\mathbf{P}^x\mathbf{X}). \tag{27}$$

$$\mathbf{B} = f(\mathbf{Y}) = \mathrm{sgn}(\mathbf{P}^y\mathbf{Y}). \tag{28}$$

From (7), we know hash codes $\mathbf{B}$ is quantified by latent semantic representations $\mathbf{V}$; then, we can obtain

$$\mathbf{V} \approx \mathbf{P}^x\mathbf{X} \approx \mathbf{P}^x\mathbf{Y}. \tag{29}$$

Then, the objective function for learning linear projection matrices $\mathbf{P}^x$ and $\mathbf{P}^y$ can be defined as

$$\min_{\mathbf{P}^x} \|\mathbf{V} - \mathbf{P}^x\mathbf{X}\|_F^2 + \gamma R(\mathbf{P}^x). \tag{30}$$

$$\min_{\mathbf{P}^y} \|\mathbf{V} - \mathbf{P}^y\mathbf{Y}\|_F^2 + \gamma R(\mathbf{P}^y). \tag{31}$$

Letting the partial derivative of (30) and (31) with respect to $\mathbf{P}^x$ and $\mathbf{P}^y$ equal zero, respectively, the closed-form solutions of $\mathbf{P}^x$ and $\mathbf{P}^y$ can be obtained by

$$\mathbf{P}^x = \mathbf{V}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}\right)^{-1}. \tag{32}$$

$$\mathbf{P}^y = \mathbf{V}\mathbf{Y}^T\left(\mathbf{Y}\mathbf{Y}^T + \gamma\mathbf{I}\right)^{-1}. \tag{33}$$

Through $\mathbf{P}^x$ and $\mathbf{P}^y$, hash codes of query data can be directly obtained. The whole querying process of JIMFH is summarized in Algorithm 1.

3.9. Time and space complexity analysis

In this subsection, we analyze the time and space complexity of JIMFH. The training time complexity of JIMFH includes hash code learning and hash function learning. Typically, learning hash codes by the iterative algorithm requires $O(m^2nT)$, where $m = \max\{d, k\}$, $d = \max\{d_x, d_y\}$, T is the iteration number, and n is the training set size. Learning hash functions requires $O(m^2n)$. Therefore, the time complexity of training JIMFH is linear to n, which is scalable for large-scale data sets. In the querying stage, the time complexity is $O(dk)$, which is extremely efficient.

The space complexity of JIMFH is determined by the space used to store intermediate variables for computing the objective function. Thus, the space complexity of JIMFH mainly lies in all the matrix variables and is $O(mn)$. Therefore, the space complexity of JIMFH is linear to n, which is acceptable in large-scale retrieval tasks.

3.10. Multiple modalities extension

The proposed JIMFH can be easily extended to cases with more modalities. Assume the training instances consist of m modalities, denoted as $\mathbf{X}^t$, $t = 1, 2, \cdots, m$. The multiple modalities extension of JIMFH in (4) can be formulated as

$$\begin{aligned}\min_{\mathbf{U}_u^t, \mathbf{U}^t, \mathbf{V}_u, \mathbf{V}_t, \mathbf{R}_t} \mathcal{F} = &\sum_{t=1}^{m} \lambda_t \left(\left\| \mathbf{X}^t - \mathbf{U}_u^t\mathbf{V}_u \right\|_F^2 + \left\| \mathbf{X}^t - \mathbf{U}^t\mathbf{V}_t \right\|_F^2 \right) + \gamma \sum_{t=2}^{m} R(\mathbf{R}_t) \\ &+ \mu \sum_{t=2}^{m} \|\mathbf{V}_1 - \mathbf{R}_t\mathbf{V}_t\|_F^2 + \gamma \sum_t R\left(\mathbf{U}_u^t, \mathbf{U}^t, \mathbf{V}_t\right) + \gamma R(\mathbf{V}_u) \\ &s.t. \sum_t \lambda_t = 1, k_u + k_s = k.\end{aligned} \tag{34}$$

where $\mathbf{U}_u^t \in \mathbb{R}^{d_t \times k_u}$ and $\mathbf{U}^t \in \mathbb{R}^{d_t \times k_s}$ are the latent vector matrices of the tth modality, $\mathbf{V}_t \in \mathbb{R}^{k_s \times n}$ is the specific representation matrix of the tth modality, $\mathbf{V}_u \in \mathbb{R}^{k_u \times n}$ is the unified representation matrix of all the modalities, $\mathbf{R}_t$ is the correlation matrix between the first modality and the tth modality ($t \geq 2$), λ^t is a nonnegative parameter that controls the weight of the tth modality, μ and γ are nonnegative parameters controlling the contributions of the corresponding terms, k_u and k_s are the dimensionalities of unified latent semantic space and specific latent semantic space, respectively, k is the hash code length, d_t is the dimensionality of the original feature space of the tth modality, and n is the number of training data.

The new optimization problem (34) can be solved directly by adjusting Algorithm 1. The closed-form solutions of $\mathbf{U}_u^t$, $\mathbf{V}_u$, $\mathbf{U}^t$, $\mathbf{V}_t$, $\mathbf{R}_t$ can be obtained by adjusting (9), (12), (13), (15) and (18), respectively.

4. Experiments

A series of cross-modal retrieval experiments are carried out on three benchmark data sets to evaluate the performance of JIMFH. In the experiments, each data set contains two modalities: image and text. Therefore, we use images to search for relevant texts that serve as image-query-text tasks. We use text to search for relevant images that serve as text-query-image tasks. In addition, the efficiency and convergence properties of JIMFH as well as its parameter sensitivity are also studied in the experiments.

4.1. Data sets

MIRFlickr[1]: The MIRFlickr data set consists 25,000 images with textural tags downloaded from Flickr. Image-tag pair is manually labeled with one or more of the 24 semantic concepts. By removing images without labels or tags, we obtain 16,738 image-tag pairs and select 2000 pairs as the test set by random selection, and the remaining pairs are served as the training set. Each image is converted to a 150-dimensional edge histogram feature vector. And its corresponding text is described as a 500-dimensional feature vector extracted by performing PCA on its index vector. For a query, pairs sharing at least one label with it are defined as its true semantic neighbors.

NUS-WIDE[2]: The NUS-WIDE data set is a real-world web image data set, which consists of 269,648 images and the associated textural tags from Flickr. Each image-tag pair is manually labeled with one or more of the 81 semantic concepts. By selecting top 10 common concepts, we obtain 186,577 image-tag pairs, from which 2000 pairs are randomly selected as the test set, and the remaining pairs are served as the training set. For each instance, the image is converted to a 500-dimensional SIFT feature of bag-of-visual-words and the text is described as an index vector of the most frequent 1000 tags. For a query, pairs sharing at least one label with it are defined as its true semantic neighbors.

MSCOCO[3]: The MSCOCO data set contains more than 300,000 real-world images and their textural captions. Each image-caption pair is manually labeled with one or more of the 80 semantic concepts. By removing images without captions or labels, 122,218 image-caption pairs can be obtained and from which 2000 pairs are randomly selected as the test set, and the remaining pairs are served as the training set. For each pair, the image is first converted to a 4,096-dimensional deep feature produced by VGG Net and then the feature is reduced to a 512-dimensional feature by PCA. The caption is first represented as a 20,629-dimensional bag-of-words vector by natural language toolkit (NLTK) and then the feature is reduced to a 512-dimensional feature by PCA. For a query, pairs sharing at least one label with it are defined as its true semantic neighbors.

4.2. Evaluation metrics

Two widely used evaluation metrics: mean Average Precision (mAP) and topN-precision are chosen to evaluate the retrieval performance of JIMFH. mAP and topN-precision are defined as follows:

mAP: Given a list of Q query instances, its mAP is defined as

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{N}\sum_{r=1}^{R}P_q(r)\delta_q(r)\ , \tag{35}$$

where N is the number of the related instances in retrieved list, $P_q(r)$ is the precision of the top-r retrieved instances of the qth query, and $\delta_q(r)$ indicates whether the rth instance is relevant to the qth query. If relevant, $\delta(r) = 1$, otherwise $\delta(r) = 0$. We set R to 100 in the experiments.

topN-precision: It reflects the variations in precision with the number of returned instances varies.

In general, a larger mAP or topN-precision value indicates superior retrieval performance.

4.3. Comparison methods

In the experiments, the following seven unsupervised multimodal hashing methods are selected as baselines to compare with JIMFH:

- **CVH** [17]: Cross-view hashing learns modal-specific hash codes by canonical correlation analysis.
- **IMH** [18]: Inter-media hashing learns modal-specific hash codes by constructing inter-modality and intra-modality similarity graphs.
- **LCMH** [19]: Linear cross-modal hashing learns modal-specific hash codes by constructing approximate similarity graphs.
- **CMFH** [21]: Collective matrix factorization hashing learns unified hash codes by collective matrix factorization.
- **STMH** [22]: Semantic topic multimodal hashing learns modal-specific hash codes by matrix factorization and topic model for image and text, respectively.
- **LSSH** [23]: Latent semantic sparse hashing learns modal-specific hash codes by sparse coding for image and matrix factorization for text.
- **RFDH** [24]: Robust and flexible discrete hashing learns unified hash codes by robust discrete matrix factorization.

Among all the comparison methods, CVH, IMH and LCMH are graph-based methods, other methods are matrix factorization-based methods. CMFH and RFDH learn unified hash codes and others learn modal-specific hash codes.

4.4. Implementation details

JIMFH has four parameters: λ, μ, γ, and k_u in objective function (4). λ controls the weights of image and text. μ controls the contribution of the correlation matching term. γ controls the contribution of the regularization term. k_u is the length of unified hash codes. In the following experiments, we set $\lambda = 0.5$, $\mu = 0.001$, and $\gamma = 0.0001$. And k_u is set to one fourth of the total length of hash code. The sensitive of JIMFH to these parameters is presented on Section 4.6.

The implementation codes of all the baselines are download from the authors's websites and they are all implemented by MATLAB. Therefore, we also implemented the proposed JIMFH by MATLAB. Parameters of comparison methods are tuned in terms of their literatures. For all the iteration methods and JIMFH, the maximum number of iteration is set to 100 and the convergence threshold is set to 0.01 in the following experiments.

4.5. Experimental results

In this section, we compare JIMFH with baselines by varying hash code length in the range of {16, 32, 64, 128} on three

[1] http://press.liacs.nl/mirflickr/mirdownload.html.
[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.
[3] http://mscoco.org/home/.

Table 1
mAP comparison on different data sets.

Task	Method	MIRFlickr				NUS-WIDE				MSCOCO			
		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 query Text	CVH	0.6444	0.6312	0.6137	0.6020	0.5216	0.5146	0.4912	0.4600	0.7288	0.8032	0.8404	0.8385
	LCMH	0.5465	0.5470	0.5775	0.5901	0.3847	0.3629	0.3678	0.3609	0.3395	0.3341	0.3723	0.4251
	IMH	0.6152	0.6196	0.6100	0.6036	0.4930	0.4852	0.4702	0.4436	0.6732	0.7756	0.8005	0.7928
	CMFH	0.6438	0.6398	0.6398	0.6168	0.4258	0.4584	0.4609	0.4611	0.7299	0.8003	0.8418	0.8661
	LSSH	0.6132	0.6294	0.6501	0.6458	0.4701	0.4357	0.4712	0.4763	0.5549	0.6489	0.7047	0.7413
	RFDH	0.6421	0.6501	0.6491	0.6541	0.4679	0.4903	0.5046	0.5130	0.7575	0.8252	0.8527	0.8656
	STMH	0.6206	0.6115	0.6313	0.5786	0.4858	0.5601	0.5593	0.5650	0.7150	0.7957	0.8459	0.8685
	JIMFH	0.6447	0.6637	0.6601	0.6570	0.5514	0.5760	0.5673	0.5725	0.7161	0.7764	0.8250	0.8335
Text query Image	CVH	0.6499	0.6355	0.6191	0.6061	0.5462	0.5310	0.4968	0.4617	0.5152	0.5671	0.5924	0.6216
	LCMH	0.5389	0.5507	0.5638	0.5784	0.3731	0.3672	0.3743	0.3802	0.3259	0.3420	0.3563	0.2966
	IMH	0.6394	0.6340	0.6236	0.6081	0.4980	0.4953	0.4746	0.4471	0.5084	0.6104	0.6040	0.6564
	CMFH	0.7058	0.7260	0.7486	0.7627	0.5049	0.5533	0.5788	0.5861	0.6395	0.6574	0.5957	0.6214
	LSSH	0.6972	0.7234	0.7470	0.7547	0.6648	0.6754	0.6904	0.6972	0.4132	0.4674	0.6001	0.6278
	RFDH	0.7009	0.7336	0.7292	0.7349	0.5300	0.5776	0.6025	0.6293	0.5543	0.5577	0.5995	0.6195
	STMH	0.7010	0.7349	0.7469	0.7539	0.6559	0.6842	0.6985	0.7066	0.5026	0.5492	0.6588	0.6594
	JIMFH	0.7078	0.7366	0.7527	0.7648	0.6656	0.6946	0.7092	0.7221	0.6647	0.6913	0.7140	0.7007

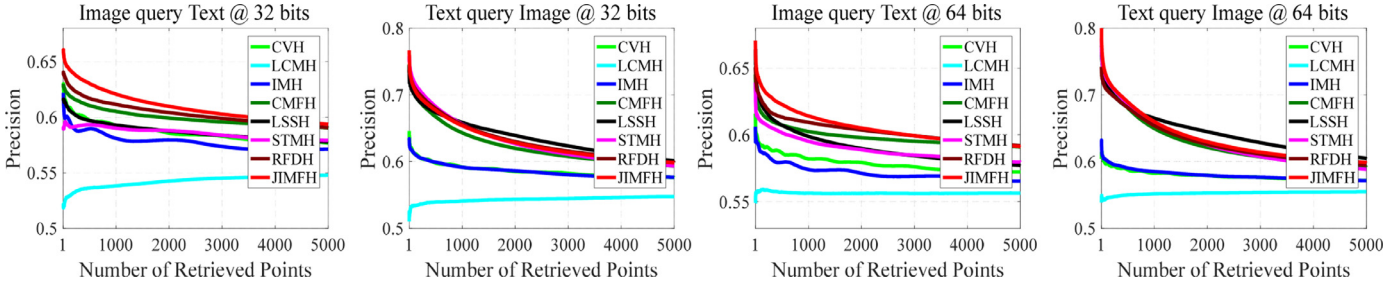

Fig. 2. topN-precision on MIRFlickr with different hash code length.

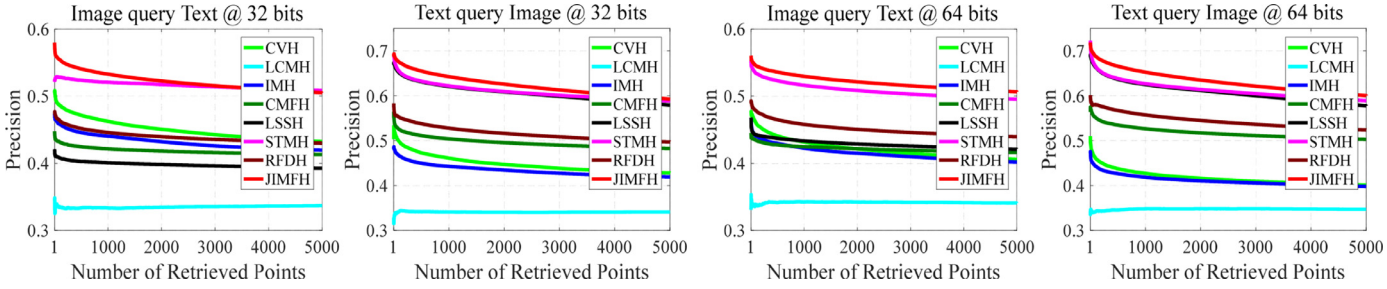

Fig. 3. topN-precision on NUS-WIDE with different hash code length.

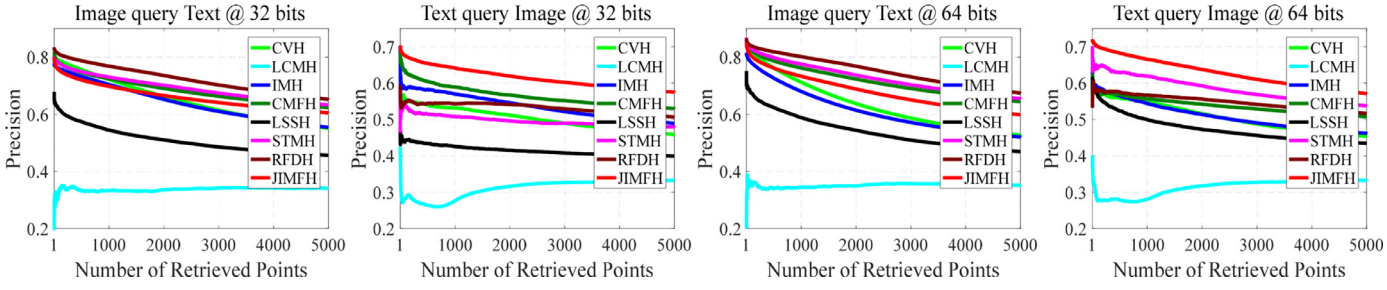

Fig. 4. topN-precision on MSCOCO with different hash code length.

data sets. Table 1 summarizes the mAP values of all the methods. Figs. 2–4 show the topN-precision curves.

MIRFlickr: Table 1 lists the mAP values of baselines and JIMFH on the MIRFlickr data set. JIMFH always achieves the best mAP values with different hash code lengths. We further observe that, the matrix factorization-based methods CMFH, LSSH, RFDH, STMH, and JIMFH outperform the graph-based methods CVH, LCMH, and IMH on both tasks. This observation shows the effectiveness of learning a shared latent semantic space through matrix factorization for multimodal hashing methods. Among matrix factorization-based methods, JIMFH provides the best performance, which shows that simultaneously learning unified and individual hash codes for multimodal data can enhance the retrieval performance. Fig. 2 shows topN-precision with 32 bit and 64 bit code lengths on the MIRFlickr data set. It can be seen that JIMFH outperforms others on the image-query-text task and performs better than others on top

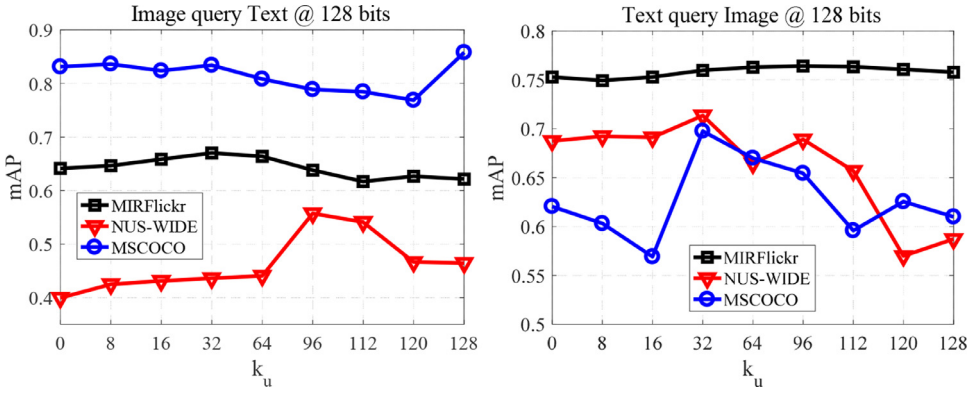

Fig. 5. mAP variations with parameter k_u.

retrieved instances on text-query-image tasks. Generally, in the retrieved list, users are more concerned with front instances. Therefore, top retrieved instances should have high accuracy. In this sense, JIMFH achieves superior performance on both retrieval tasks.

NUS-WIDE: Table 1 and Fig. 3 show mAP values and topN-precision curves on the NUS-WIDE data set. We can see that 1) JIMFH is always superior to others in terms of both mAP values and topN-precisions. 2) JIMFH has a substantial improvement of 10% over the most related baseline CMFH on both tasks. This shows the effectiveness of retaining specific properties of multimodal data on their hash codes. 3) The trend among all the methods is that longer hash codes often yield better performance. This is because longer hash codes can retain more information. It can be further observed that JIMFH with the minimum code length (16 bit) performs better than many methods with the maximum code length (128 bit) on the image-query-text task. These observations reflect the effectiveness of JIMFH on the NUS-WIDE data set.

MSCOCO: The mAP values of all the retrieved results on the MSCOCO data set are listed in Table 1. It is clear that JIMFH increases other results by approximately 4% on the text-query-image task. On the image-query-text task, RFDH performs better than the others. Although RFDH performs better than JIMFH on the image-query-text task, its mAP value is not as good on the text-query-image task and approximately 11% lower than JIMFH. Compared with most related baseline CMFH, JIMFH performs much better on the text-query-image task with approximately 6% improvement and underperforms slightly on the image-query-text task with an approximately 2% decrease. In general, we can observe that although JIMFH does not obtain the best mAP values on the image-query-text task, its performance is still good as its mAP is only slightly lower than the best. However, it achieves much better results on text-query-image tasks. Therefore, JIMFH can achieve promising performance on the MSCOCO data set. Fig. 4 shows topN-precision curves on the MSCOCO data set. Its trend is consistent with the mAP results. Again, JIMFH yields comparable accuracy on the image-query-text task and beats all other baselines on the text-query-image task.

4.6. Comprehensive analysis

4.6.1. Contribution degree of JMF and IMF

In this subsection, we analyze the effect of coding length of JMF and IMF. In JIMFH, the coding length of JMF is k_u and IMF is $k_s = k - k_u$. In the experiment, the whole hash code length k is set to 128, k_u is increasing from 0 to 128, and consequently k_s is decreasing from 128 to 0 with k_u increasing. When $k_u = 0$ and $k_u = 128$, JIMFH only includes the JMF alone and the IMF alone, respectively. Fig. 5 shows the mAP values of JIMFH with varying k_u on all three data sets. It is clear that muting either the JMF or IMF can substantially degrade the performance. The only exception is the image-query-text task on the MSCOCO data set, where JIMFH only includes the JMF and performs best. This phenomenon is consistent with that in Table 1, where CMFH performs slightly better than JIMFH on the image-query-text task on the MSCOCO data set. However, JIMFH with both part performs much better in most cases, especially on text-query-image tasks on the NUS-WIDE and MSCOCO data sets. Therefore, we conclude that both terms are critical to cross-modal retrieval results.

Table 2
Training time (in seconds) comparisons.

Method/ Size of data set	1500	2000	3000	5000	10000
CVH	5	6	6	6	7
LCMH	5	6	8	13	25
IMH	18	35	108	628	2799
CMFH	16	19	25	40	67
LSSH	76	80	89	100	142
STMH	12	15	21	31	51
RFDH	67	80	122	238	488
JIMFH	32	44	63	92	112

4.6.2. Convergence analysis

The convergence of Algorithm 1 has been proven theoretically in Section 3.7. In this section, we experimentally validate its convergence on all three data sets. The convergence curves of JIMFH are illustrated in Fig. 6. Each point records the objective value of (4) at each iteration. Obviously, as the iteration number increases, the objective value monotonically decreases and tends to be stable after several iterations. In general, JIMFH can converge in fewer than 10 iterations. This reflects the efficiency of JIMFH.

4.6.3. Training time analysis

To further verify the efficiency of JIMFH, we conduct experiments on the NUS-WIDE data set to compare the training times of the baselines and JIMFH. In the experiments, the hash code length is fixed at 32 bits, and the training set size is increasing from 1500 to 10,000. The training time of all the methods under the same settings is listed in Table 2. We can see that IMH and RFDH take more time than other methods for training the model. CVH takes only a few seconds for training. This is because it dose not need iteration. Other methods, including JIMFH, need dozens of seconds for training. Generally, JIMFH can train the model efficiently.

4.6.4. Parameter sensitivity analysis

In this section, we perform an experimental analysis of parameter sensitivities. We fix hash code length at 32. When analyz-

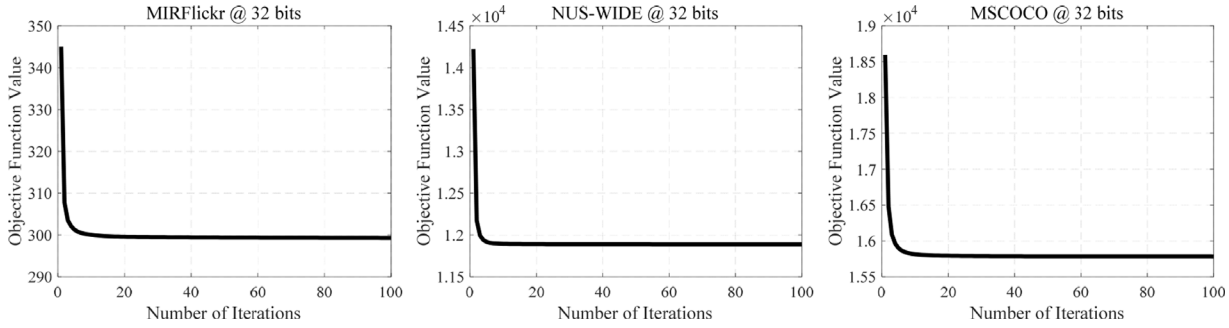

Fig. 6. Convergence study of JIMFH.

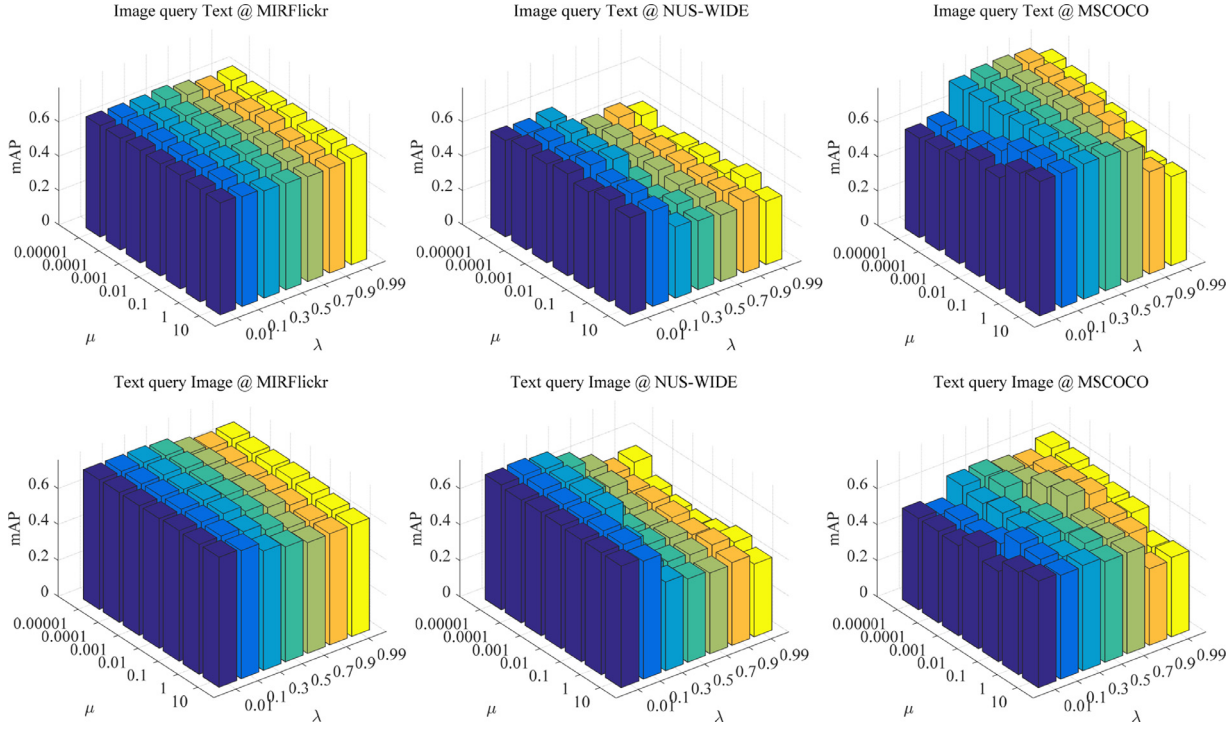

Fig. 7. mAP variations with parameters λ and μ.

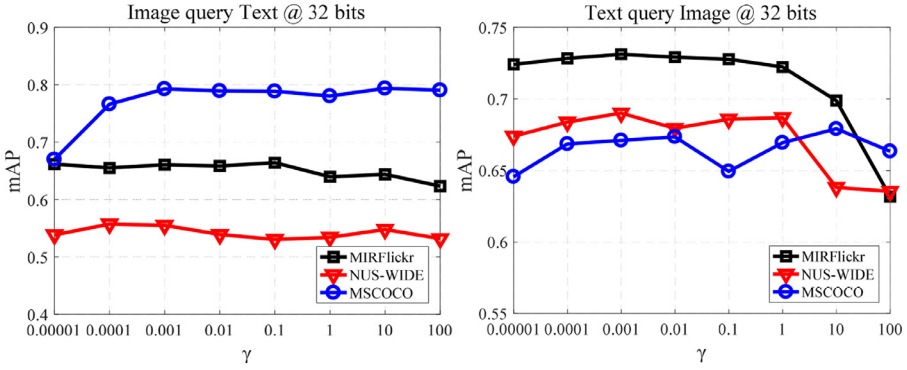

Fig. 8. mAP variations with parameter γ.

ing one parameter, we vary its value while fixing the values of other parameters. There are four main parameters in JIMFH: k_u is the length of the unified hash code, λ controls the weights of the image and text, μ controls the contribution of the correlation matching term, and γ controls the contribution of the regularization term. We analyzed the sensitivity of k_u in the previous section. In this section, we mainly analyze the sensitivity of λ, μ and γ.

The mAP values of JIMFH with varying λ and μ on all three data sets are shown in Fig. 7. It can be noticed that JIMFH usually achieves superior performance on all the data sets with μ being small (e.g., less than 0.01). JIMFH achieves better performance when λ is small on the NUS-WIDE data set and large on the MSCOCO data set. This is reasonable because λ leverages the importance of image and text. On the MSCOCO data set, the deep image feature is more effective than the text feature; therefore, a

large λ can bring better results. On the NUS-WIDE data set, the text feature is more effective; therefore, a small λ can bring better results. Generally, the proposed JIMFH is insensitive to both λ and μ. It achieves good performance in a wide range of λ and μ on all the data sets.

Fig. 8 shows the mAP values of JIMFH with varying γ. In general, it can be observed that when γ is greater than 1, the performance of JIMFH decreases rapidly. When γ is too small, the performance of JIMFH is also less desirable. Usually, JIMFH performs well when γ is in the range of [0.0001, 0.1].

5. Conclusion

In this paper, a joint and individual matrix factorization hashing (JIMFH) is proposed for large-scale cross-modal retrieval. It jointly learns an optimal combination of unified hash codes and individual hash codes for multimodal data, in which it not only learns individual binary codes for each modality to retain its specific property but also learns unified binary codes for different modalities to preserve their shared properties. An effective optimization algorithm is proposed to solve the proposed JIMFH, and the theoretical analysis is also given in detail. Extensive experiments on three benchmark multimodal data sets highlight the superiority of JIMFH over the state-of-the-art on large-scale cross-modal retrieval applications.

Recently, deep hashing has gained wide attention and can achieve promising result. Comparing with unsupervised deep multimodal hashing methods, the retrieval precision of the proposed JIMFH is slightly lower. However, JIMFH has lower computational complexity and less training time. In the future work, we will incorporate the joint and individual matrix factorization idea of JIMFH into deep matrix factorization models to further improve the hashing performance.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this work.

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61702394, 61772402, 61972302, 61876146, 61802294 and 61702400, in part by the China Postdoctoral Science Foundation funded project under Grants 2018T111021 and 2017M613082, in part by the National Key Research and Development Program of China under Grant 2016QY01W0200, in part by the Scientific Research Project of Shaanxi Province, China under Grants 2019JQ-205, 2019ZDLGY13-01, 2019ZDLGY13-07 and 2017ZDXM-GY-002, in part by the Science and Technology Project of Xi'an, China under Grant 201809170CX11JC12, and in part by the Fundamental Research Funds for the Central Universities under Grant XJS190306.

References

[1] V.E. Liong, J. Lu, Y.-P. Tan, Cross-modal discrete hashing, Pattern Recognit. 79 (2018) 114–129.
[2] X. Shen, W. Liu, I.W. Tsang, Q.-S. Sun, Y.-S. Ong, Multilabel prediction via cross-view search, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4324–4338.
[3] X. Shen, F. Shen, L. Liu, Y.-H. Yuan, W. Liu, Q.-S. Sun, Multiview discrete hashing for scalable multimedia search, ACM Trans. Intell. Syst. Technol. 9 (5) (2018) 1–21.
[4] F. Zhong, Z. Chen, G. Min, Deep discrete cross-modal hashing for cross-media retrieval, Pattern Recognit. 83 (2018) 64–77.
[5] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, D. Tao, Unsupervised semantic-preserving adversarial hashing for image search, IEEE Trans. Image Process. 28 (8) (2019) 4032–4044.
[6] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, R. Hong, Self-supervised video hashing with hierarchical binary auto-encoder, IEEE Trans. Image Process. 27 (7) (2018) 3210–3221.
[7] J. Wang, T. Zhang, N. Sebe, H.T. Shen, et al., A survey on learning to hash, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 769–790.
[8] D. Zhang, W. Li, Large-scale supervised multimodal hashing with semantic correlation maximization, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014, pp. 2177–2183.
[9] Z. Lin, G. Ding, M. Hu, J. Wang, Semantics-preserving hashing for cross-view retrieval, in: Proceedings of the 28th International Conference on Computer Vision and Pattern Recognition, 2015, pp. 3864–3872.
[10] K. Ding, B. Fan, C. Huo, S. Xiang, C. Pan, Cross-modal hashing via rank-order preserving, IEEE Trans Multimed. 19 (3) (2017) 571–585.
[11] D. Wang, X. Gao, X. Wang, L. He, B. Yuan, Multimodal discriminative binary embedding for large-scale cross-modal retrieval, IEEE Trans. Image Process. 25 (10) (2016) 4540–4554.
[12] X. Xu, F. Shen, Y. Yang, H.T. Shen, X. Li, Learning discriminative binary codes for large-scale cross-modal retrieval, IEEE Trans. Image Process. 26 (5) (2017) 2494–2507.
[13] D. Wang, X. Gao, X. Wang, L. He, Label consistent matrix factorization hashing for large-scale cross-modal similarity search, IEEE Trans. Pattern Anal. Mach. Intell. 41 (10) (2019) 2466–2479.
[14] Q. Jiang, W. Li, Deep cross-modal hashing, in: Proceedings of the 30th International Conference on Computer Vision and Pattern Recognition, 2017, pp. 3233–3240.
[15] C. Li, C. Deng, N. Li, W. Liu, X. Gao, D. Tao, Self-supervised adversarial hashing networks for cross-modal retrieval, in: Proceedings of the 31st International Conference on Computer Vision and Pattern Recognition, 2018, pp. 4242–4251.
[16] C. Yan, X. Bai, S. Wang, J. Zhou, E.R. Hancock, Cross-modal hashing with semantic deep embedding, Neurocomputing 337 (2019) 58–66.
[17] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011, pp. 1360–1367.
[18] 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 19th ACM SIGMOD International Conference on Management of Data, 2013, pp. 785–796.
[19] X. Zhu, Z. Huang, H.T. Shen, X. Zhao, Linear cross-modal hashing for efficient multimedia search, in: Proceedings of the 21st ACM International Conference on Multimedia, 2013, pp. 143–152.
[20] X. Shen, F. Shen, Q.S. Sun, Y. Yang, Y.H. Yuan, H.T. Shen, Semi-paired discrete hashing: learning latent hash codes for semi-paired cross-view retrieval, IEEE Trans. Cybern. 47 (12) (2017) 4275–4288.
[21] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: Proceedings of the 27th International Conference on Computer Vision and Pattern Recognition, 2014, pp. 2083–2090.
[22] D. Wang, X. Gao, X. Wang, L. He, Semantic topic multimodal hashing for cross–media retrieval, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 3890–3896.
[23] 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, 2014, pp. 415–424.
[24] D. Wang, Q. Wang, X. Gao, Robust and flexible discrete hashing for cross–modal similarity search, IEEE Trans. Circuits Syst. Video Technol. 28 (10) (2018) 2703–2715.
[25] D. Hu, F. Nie, X. Li, Deep binary reconstruction for cross-modal hashing, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 1398–1406.
[26] J. Zhang, Y. Peng, M. Yuan, Unsupervised generative adversarial cross-modal hashing, in: Proceedings of the 31th AAAI Conference on Artificial Intelligence, 2018.
[27] E. Yang, C. Deng, T. Liu, W. Liu, D. Tao, Semantic structure-based unsupervised deep hashing, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 1064–1070.
[28] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G.R. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: Proceedings of the 18th ACM International Conference on Multimedia, 2010, pp. 251–260.
[29] F. Zhu, L. Shao, M. Yu, Cross-modality submodular dictionary learning for information retrieval, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 1479–1488.
[30] Y.T. Zhuang, Y.F. Wang, F. Wu, Y. Zhang, W.M. Lu, Supervised coupled dictionary learning with group structures for multi-modal retrieval, in: Proceedings of the 27th AAAI Conference on Artificial Intelligence, 2013.
[31] A. Sharma, A. Kumar, H. Daume, D.W. Jacobs, Generalized multiview analysis: a discriminative latent space, in: Proceedings of the 25th IEEE International Conference on Computer Vision and Pattern Recognition, 2012, pp. 2160–2167.
[32] Y. Jia, M. Salzmann, T. Darrell, Learning cross-modality similarity for multinomial data, in: Proceedings of the 13th International Conference on Computer Vision, 2011, pp. 2407–2414.
[33] Y. Wang, F. Wu, J. Song, X. Li, Y. Zhuang, Multi-modal mutual topic reinforce modeling for cross-media retrieval, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 307–316.

[34] W. Wang, X. Yang, B.C. Ooi, D. Zhang, Y. Zhuang, Effective deep learning-based multi-modal retrieval, VLDB J. 25 (1) (2016) 79–101.
[35] C. Wang, H. Yang, C. Meinel, Deep semantic mapping for cross-modal retrieval, in: Proceedings of the 27th International Conference on Tools with Artificial Intelligence, 2015, pp. 234–241.
[36] J. Tang, Z. Li, Weakly supervised multimodal hashing for scalable social image retrieval, IEEE Trans. Circuits Syst. Video Technol. 28 (10) (2017) 2730–2741.
[37] J. Zhang, Y. Peng, M. Yuan, SCH-GAN: semi-supervised cross-modal hashing by generative adversarial network, IEEE Trans. Cybern. (2018).
[38] A.P. Singh, G.J. Gordon, Relational learning via collective matrix factorization, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 650–658.
[39] D. Wang, X. Gao, X. Wang, Semi-supervised nonnegative matrix factorization via constraint propagation, IEEE Trans. Cybern. 46 (1) (2016) 233–244.
[40] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (1) (2010) 19–60.

Di Wang received the Ph.D. degree in intelligent information processing from Xidian University, Xi'an, China, in 2016. She is currently an associate professor with the School of Computer Science and Technology, Xidian University. Her research interests include machine learning and multimedia information retrieval.

Quan Wang received the B.Sc., M.Sc., and Ph.D. degrees in computer science and technology from Xidian University, Xi'an, China. He is currently a Professor with the School of Computer Science and Technology, Xidian University. His current research interests include input and output technologies and systems, image processing, and image understanding.

Lihuo He received the B.Sc. degree in electronic and information engineering and Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2008 and 2013. He is currently an associate professor with Xidian University. His research interests focus on computational vision, pattern recognition and artificial intelligence.

Xinbo Gao received the B.Eng., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow at the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a postdoctoral research fellow at the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong Professor of Ministry of Education of PR China, a Professor of Pattern Recognition and Intelligent System, and the Dean of Graduate School of Xidian University. His current research interests include Image processing, computer vision, multimedia analysis, machine learning and pattern recognition. He has published six books and around 300 technical articles in refereed journals and proceedings. Prof. Gao is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as the General Chair/Co-Chair, Program Committee Chair/Co-Chair, or PC Member for around 30 major international conferences. He is a Fellow of the Institute of Engineering and Technology and a Fellow of the Chinese Institute of Electronics.

Yumin Tian received the B.Sc., and M.Sc. degrees in computer application from Xidian University, China, in 1984, and 1987, respectively. She is currently a Professor in the School of Computer Science and Technology, Xidian University. Her research interests include image processing, 3D shape recovery, digital watermarking, and computer vision.