8-2016

Unsupervised multi-graph cross-modal hashing for large-scale multimedia retrieval

Liang XIE
Wuhan University of Technology

Lei ZHU
Singapore Management University, lzhu@smu.edu.sg

Guoqi CHEN
Wuhan University of Technology

Published in Multimedia Tools and Applications, August 2016, Volume 75, Issue 15, pp 9185–9204

https://doi.org/10.1007/s11042-016-3432-0

Accepted version

Unsupervised multi-graph cross-modal hashing for large-scale multimedia retrieval

Liang Xie, Wuhan University of Technology

Lei Zhu, Singapore Management University

Guoqi Chen, Wuhan University of Technology

Abstract

With the advance of internet and multimedia technologies, large-scale multi-modal representation techniques such as cross-modal hashing, are increasingly demanded for multimedia retrieval. In cross-modal hashing, three essential problems should be seriously considered. The first is that effective cross-modal relationship should be learned from training data with scarce label information. The second is that appropriate weights should be assigned for different modalities to reflect their importance. The last is the scalability of training process which is usually ignored by previous methods. In this paper, we propose Multi-graph Cross-modal Hashing (MGCMH) by comprehensively considering these three points. MGCMH is unsupervised method which integrates multi-graph learning and hash function learning into a joint framework, to learn unified hash space for all modalities. In MGCMH, different modalities are assigned with proper weights for the generation of multi-graph and hash codes respectively. As a result, more precise cross-modal relationship can be preserved in the hash space. Then Nyström approximation approach is leveraged to efficiently construct the graphs. Finally an alternating learning algorithm is proposed to jointly optimize the modality weights, hash codes and functions. Experiments conducted on two real-world multi-modal datasets demonstrate the effectiveness of our method, in comparison with several representative cross-modal hashing methods.

Keywords

Cross-modal hashing, Multi-graph learning, Cross-media retrieval

1 Introduction

In recent years, there has been an explosion in the scale of multimedia data on the web. For example, Flickr hosts billions of images, and it has more than 3.5 million new images uploaded daily. Traditional representation or feature learning methods, including bag-of-visual-words (BoVW) [7], Fisher Vector (FV) [23], Sparse Coding [22] and Dictionary Learning [41], cannot work well and even be computationally intractable for the retrieval of large-scale multimedia data.

When designing an efficient representation technique for large-scale multimedia data, both computational cost and memory cost should be considered. Hashing technology, which is a representative binary representation learning approach, has gained much attention recently. Its core idea is to learn compact binary codes to represent high-dimensional data by hash functions. On one hand, these binary codes, which preserve the neighborhood relationships of original data, occupy a small amount of memory space. Thus a large number of codes can be stored in RAM. On the other hand, based on the Hamming distance between binary codes, fast search can be easily implemented by simple but efficient XOR and bit-count operations.

Existing hashing methods can be generally divided into two categories [27]. The first is random projection based hashing methods. Locality Sensitive Hashing (LSH) [1] is one of the most representative methods of this kind, it uses several hash functions consisting of random linear projection. The disadvantage of LSH is that it may lead to quite inefficient (long) codes in practice, because the hash functions of LSH is data-independent [45]. The other category is machine learning based hashing methods, which can learn effective data-dependent hash functions. Therefore, many machine learning methods are designed for hashing, such as Spectral Hashing [35], Self-taught Hashing (STH) [45], K-means Hashing [13], PCA Hashing [30] and Anchor Graph Hashing [18].

Although the above mentioned hashing techniques can achieve promising retrieval performance, most of them are only applied to unimodal data. With the advance of internet and multimedia technologies, large amount of multi-modal data are generated, shared and accessed on social websites, e.g., Flickr, Wikipedia and YouTube. Images or videos on the web are usually associated with text information, such as textual tags or comments. Traditional unimodal hashing methods cannot work well in the multi-modal scenario. Recently, several cross-modal hashing methods are proposed, including CMSSH [54], CVH [15], MLBE [50], IMH [27] and THH [53]. Most of them leverage machine learning technologies to learn hash functions which can project different modalities into a unified space. Intuitively, machine learning is the best choice for cross-modal hashing. In common cross-modal analysis, the relationship between different modalities is unknown, and they should be learned from multi-modal data. Therefore, it is promising to learn the cross-modal relation and hash space simultaneously, and more specifically, to preserve the cross-modal relationship in the learned hash space.

Despite the capability to deal with multi-modal data, there are three essential problems which should be seriously considered in designing an effective and practical hashing method. At first, training data is usually unlabeled, and manually labeling training data is time-consuming and expensive. Therefore, it is more practical to learn hash functions in the unsupervised manner. The second is that the weights of different modalities should be considered for hashing. Generally, different modalities may have different contributions to the cross-modal relation and hash functions. Traditional cross-modal methods [15, 36] treat each modality equally, thus if one modality is very noisy, their performance may be significantly affected. The other is the scalability of learning process. Although it is quite efficient to use hash codes for search, learning hash functions in some existing methods is not so efficient. For example, the graph-based hashing approaches, including STH and IMH, are shown to be effective, but the time complexity of their learning process is quadratic to the size of training data.

To solve the above problems, in this paper we propose a novel hashing method, which is termed as Multi-graph Cross-modal Hashing (MGCMH), for multimedia search. MGCMH is an unsupervised method which requires no label information in the training data. It is formulated in a joint mutli-graph framework, which simultaneously learns weights of modalities and their unified latent hash space. To solve the out-of-sample problem, we also learn the hash functions to project new data into this space. By integrating multi-graph learning and hash function learning, we obtain a joint framework which both learns optimal hash codes and hash functions. As a result, all modalities are mapped into the unified hash space by hash functions. Then in the training process, since graph construction is time-consuming, Nyström approach is adopted to approximate the

graph of each modality. Finally, in the optimization of hash codes and functions, an alternating learning process is proposed. The advantages of MGCMH are summarized as follows:

- Since MGCMH is unsupervised, it is suited to real-world applications where label information is usually scarce and expensive to obtain.
- In multi-graph framework, two types of weights are used for graphs and hash codes respectively. Therefore, the importance of each modality can be comprehensively considered, and the hash space constructed by MGCMH can better correlate different modalities.
- In the graph construction, Nyström approximation is used. So the training process of MGCMH is efficient, and its time complexity is linear to the size of training data.
- Experiments conducted on real-world multi-modal datasets demonstrate the better performance of MGCMH compared with several representative cross-modal hashing methods.

The rest of this paper is organized as follows. Section 2 discusses related work about cross-modal learning and hashing. In Section 3, we describe the formulation of MGCMH and its optimization, then make a discussion about it. Section 4 shows the experimental results on Wikipedia and NUS-WIDE. Finally conclusions and future work are presented in Section 5.

2 Related work

2.1 Cross-modal learning

Cross-modal learning, which is related to our work, has been widely used for various multimedia applications, such as classification and retrieval. In cross-modal (multi-modal) methods, determining the modality weight is essential for combining and correlating different modalities. Image annotation/classification is a typical application of cross-modal methods. Multiple Kernel Learning (MKL) [29] learns weights of kernels from different modalities, and it is applied to multi-modal image classification [12]. Liu et al. [17] propose multiview Hessian Regularization (mHR) for image annotation, mHR assigns kernel weights and Hessian weights to different modalities. Luo et al. [20] propose Multiview Matrix Completion (MVMC) for image classification, and they apply a cross-validation strategy to learn the modality weights. Cross-modal (multi-modal) learning is also used for video analysis, Ma et al. [21] propose Riemannian weighted Semi-Supervised Multi-feature learning (RSSM) for video action and event recognition. RSSM assigns modality weights for both Laplacian graphs and Riemannian distances, and a maximum entropy regularization is imposed to avoid trivial solution. Multi-task learning is effective for cross-modal analysis, such as FEGA-MTL [40], Multitask LDA [39] and CMMTL [36].

Another major application of cross-modal learning is multimedia retrieval [37]. In [6], two types of cross-modal relationship: correlation and abstraction are studied. It uses Canonical Correlation Analysis (CCA) [14], Kernel CCA (KCCA) [26] and Cross-modal Factor Analysis (CFA) [16] for correlation learning, and uses Logistic Regression [9], Support Vector Machine (SVM) [3] and Boosting [25] for abstraction learning. Wang et al. [32] propose to learn coupled feature spaces for cross-modal retrieval, its framework consists of the coupled linear regression and a trace norm which enforces the relevance of different modalities. Some cross-modal methods [11, 42, 43] rely on graph learning based approach. Recently, deep learning methods are becoming a popular approach for cross-modal retrieval, such as Corr-AE [10], MSAE and MDNN [34].

2.2 Unimodal hashing

Based on the usage of label information, existing hashing methods can be divided to supervised/semi-supervised and unsupervised hashing methods. Graph learning is confirmed to be effective in hashing. Anchor Graph Hashing (AGH) [18] constructs anchor graph, which is similar to Nyström approximation used in this paper. However, AGH is unimodal method, thus it is not suited to cross-modal retrieval. Discrete Graph Hashing (DGH) [19] improves AGH by using discrete optimization to directly learn binary codes, and it is also only suited to unimodal data. Both AGH and DGH are unsupervised methods, and graph learning can be applied to supervised or semi-supervised hashing, such as SSH [31]. Despite graph learning approach, other learning methods have also been extensively studied for hashing. Latent Factor Hashing (LFH) [49] is a supervised approach which learns hash codes based on latent factor model. Supervised Hashing with Pseudo Labels (SHPL) [28] uses the cluster centers as pseudo label, and then linear discriminant analysis (LDA) based trace ratio is used for hashing. Some studies [33, 51] use active learning for semi-supervised hashing, to actively select the most informative labels for hash function learning.

2.3 Cross-modal hashing

In cross-modal hashing, both supervised/semi-supervised and unsupervised methods are studied. Supervised cross-modal hashing learns the cross-modal correlation from the class labels, and different modal data with the same label are relevant. Cross-modality Similarity-sensitive Hashing (CMSSH) [2] uses supervised similarity learning to embed the input data from two modalities into the hash space. Multimodal Latent Binary Embedding (MLBE)[50] uses class labels to construct the inter-modality similarity matrices, and learns hash functions in a probabilistic framework. Semantic Correlation Maximization (SCM) [44] integrates semantic labels into the hash function learning. It uses sequential learning for non-orthogonal projection to reduce the quantization error. Supervised methods require class labels in training data, which is difficult to obtain in practice. Some supervised methods can be applied to the unsupervised case. Cross-view Hashing (CVH) [15] requires predefined affinity matrix of the training data, which is usually obtained from labels. If training data contain no labels, the affinity matrix becomes an identity matrix, and CVH becomes an unsupervised method. In [4], multi-graph learning is used for semi-supervised hashing, and its main difference to this work is that it still requires label information to learn hash functions.

Unsupervised cross-modal hashing can use training data without class labels, it exploits the co-occurrence information that different modal data in the same document is relevant, thus it is more practical than supervised hashing. Inter-media Hashing (IMH) [27] preserves both inter-media consistency and intra-media consistency, and learns hash functions by solving an eigenvalue problem. In order to preserve the intra-media consistency, IMH constructs similarity graphs, which is very time consuming when the training data is large. Collective Matrix Factorization Hashing (CMFH) [8] learns unified hash codes by the matrix factorization of each modalities. CMFH can increase the search accuracy by combining multi-modal information sources. CMSTH [38] applies self-taught learning to effectively correlate cross-modal data in hashing.

3 Multi-graph cross-modal hashing

The learning and retrieval process of Multi-Graph Cross-Modal Hashing (MGCMH) are illustrated in Fig. 1. In the learning process, Nyström approach is used to construct the approximate graph for each modality respectively, then these graphs are combined by modality weights. MGCMH uses a joint framework which consists of multi-graph learning and hash function learning. Hash codes of database and hash function of each modality are simultaneously learned by the joint framework of MGCMH. In the retrieval process, both image and text queries are supported. The hash codes of each query are first computed by the corresponding hash functions, and then they are compared with database hash codes via their hamming distance.

Fig. 1 Learning and retrieval process of MGCMH

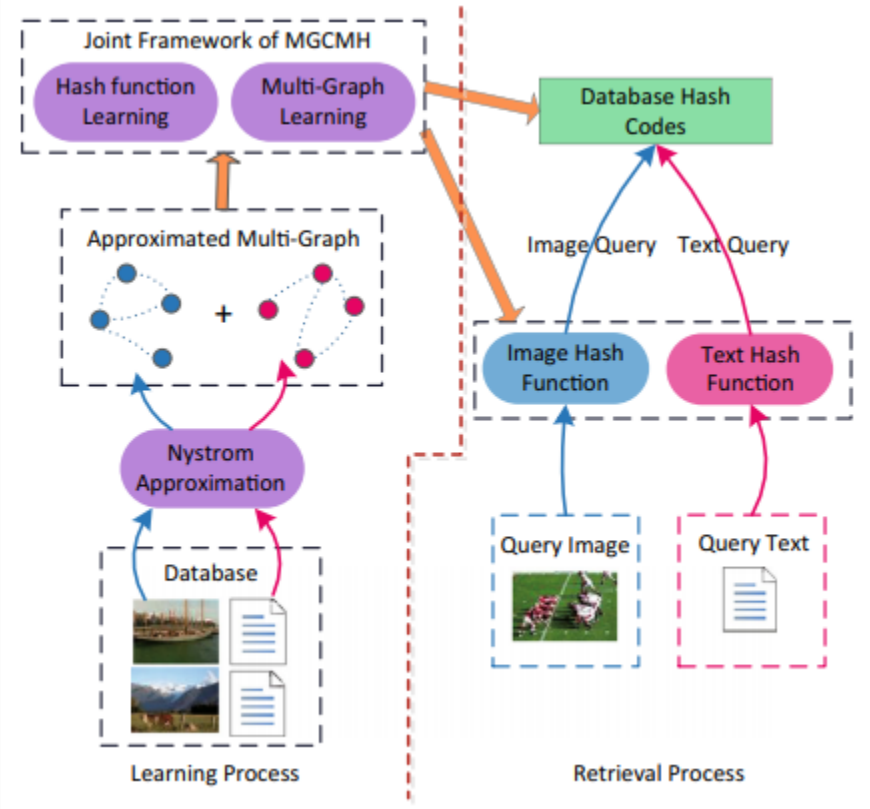


Table 1 List of notations

Notation	Description
N	Size of multi-modal training data
M	Number of modalities, $M = 2$ in this paper
K	Code length
H	Hash codes of training data
X_m	$N \times d_m$ Feature matrix of modality m
A_m	$N \times N$ Graph matrix modality m
V_m	Nyström embedding vector for approximating A_m
W_m	$d_m \times K$ Weight matrix of hash function for modality m
θ_m	Weight of modality m for multi-graph combination
α_m	Weight of modality M for generating multi-modal hash codes
β_m, γ, μ	Penalty parameters in MGCMH

3.1 Formulation of MGCMH

Suppose there are N multi-modal documents $o_n|^N_{n=1}$ for training. Each document $o = \{x_1,\ldots,x_M\}$ consists of M modalities, where x_m is the feature of the m-th modality. We only consider the condition that each document consists of an image and a text. Thus $M = 2$ in this paper, and our method can be also applied to the condition that $M > 2$. $m = 1$ denotes image modality and $m = 2$ denotes text modality. For simplicity, a list of notations used in this paper are shown in Table 1.

Our goal is to learn a set of hash codes $H=[h^T_1,\ldots,h^T_N]^T$ for all documents, where $h_n \in \{0, 1\}^{1\times K}$, K is the code length. It has been shown that optimal hash codes can be obtained when average Hamming distance between similar points is minimal [35]. Based on this principle, we formulate the following multi-graph framework to optimize the hash codes:

$$\sum_{m=1}^{M}\sum_{i,j=1}^{N}\theta_m a_{ij}^m\left(h_i-h_j\right)^2 \qquad (1)$$

where $a^m_{ij} = \exp(dis^{ij}_m / \delta_m)$, dis^{ij}_m is the m-th modal distance between the i-th and j-th documents, and δ_m is the mean of distances. We use L2 distance for images, and cosine distance for texts. θ_m is the weight of m-th modality.

This multi-graph framework has two advantages. The first is that hash codes are generated from multi-modal similarity. Principally, combining multiple modalities can preserve more semantic correlation than using single modality only. Besides, the semantic information contained in different modalities may be unbalanced. Unlike previous methods which treat all modalities equally, in our framework, the similarities of different modalities have different weights which can be automatically optimized.

We can reformulate (1) as $Tr(H^T L H)$, where $L=\sum^M_{m=1}\theta_m L_m$, $L_m = D_m - A_m$ is the Laplacian matrix for m-th modality. $D_m \in R^{N\times N}$ is the diagonal matrix, whose element $d^m_{ii} = \sum^N_{j=1} a^m_{ij}$, $A^m \in R^{N\times N}$ is the matrix whose element is a^m_{ij}.

Besides learning the hash codes H of all training documents, we also have to project new documents into this hash space. For each modality m, we adopt the projection function:

$$f_m(X_m) = X_m W_m, \quad m = 1,\ldots,M \qquad (2)$$

where $X_m \in R^{N\times d_m}$ is the feature matrix of m-th modality, $W_m \in R^{d_m\times K}$ is the projection matrix.

Finally we have the overall objective function which optimizes both H and W_m:

$$\begin{aligned}\min_{H,W_m}\ &\sum_{m=1}^{M}\theta_m\left(\|X_m W_m - H\|_F^2+\beta_m\|W_m\|_F^2\right)+\gamma Tr\left(H^T L H\right)\\ s.t.\ &H^T H = I,\\ &\sum_{m=1}^{M}\theta_m = 1\end{aligned} \qquad (3)$$

3.2 Efficient construction of graphs and hash codes

In practice, it is difficult to directly compute the similarity graphs. The computation process is time consuming, and the time complexity of the graph construction is about $O(N^2)$. If N is very large, then it is unaffordable to

spend a great deal of time (may be several days) to construct graphs. To this end, we adopt Nyström approximation [47], which is an efficient graph approximation method, to construct graphs.

Nyström method approximates its graph matrix A by:

$$A = A_{NE} A_{EE}^{-1} A_{NE}^{T} \quad (4)$$

Where $A_{EE} \in \mathrm{R}^{E \times E}$ is the sub-matrix, it is constructed by E samples which are randomly selected from N training samples. A_{NE} denotes the sub-matrix of A with E columns.

Since A_{EE} is symmetric and positive definite, its reverse can be approximated by $A^{-1}{}_{EE} = U\Lambda^{-1}U^{T}$. Where $\Lambda \in \mathrm{R}^{E' \times E'}$ is a diagonal matrix, and its diagonal elements correspond to E' largest eigenvalues of A_{EE}. $U \in \mathrm{R}^{E' \times E'}$ consists of the corresponding eigenvectors.

Then we can rewrite (4) as:

$$A = A_{NE} U \Lambda^{-1} U^{T} A_{NE}^{T} = V V^{T} \quad (5)$$

where $V = A_{NE} U\Lambda^{-1/2}$ can be regarded as the explicit embedding of training features. For m-th modality, we can first compute V_m, then A_m can be efficiently constructed.

Besides the graph construction, we have to construct the hash codes, for directly optimizing hash codes is also time-consuming. Inspired by the work in [48], we assume that H can be constructed from V_m, the difference is that we use multiple modalities for construction. H can be constructed by:

$$H = \sum_{m=1}^{M} \alpha_m V_m P_m \quad (6)$$

where $P_m \in \mathrm{R}^{Em' \times K}$ is the construction matrix of modality m, E_m' is the column dimension of V_m. α_m is the weight of modality m for constructing codes.

After the construction of graphs and hashing codes, the optimization of H is transformed to the optimization of P_m. α_m and θ_m are the weights for the construction of hash space H and Laplacian matrix L respectively. H should preserve the relationship in L, thus θ_m and α_m should be consistent. We add a penalty term to our objective function (3), then it becomes:

$$\begin{aligned} \min_{P_m, \alpha_m, \theta_m} & \sum_{m=1}^{M} \theta_m \left(\| X_m W_m - H \|_F^2 + \beta_m \| W_m \|_F^2 \right) \\ & + \gamma Tr\left(H^T L H \right) + \mu \sum_{m=1}^{M} (\alpha_m - \theta_m)^2 \\ s.t. \quad & H^T H = I, \\ & \sum_{m=1}^{M} \alpha_m = 1, \\ & \sum_{m=1}^{M} \theta_m = 1 \end{aligned} \quad (7)$$

where μ is the parameter of penalty term.

3.3 Optimization

By setting the derivative of (7) w.r.t. W_m to zero, we have:

$$W_m = \left(X_m^T X_m + \beta I_m\right)^{-1} X_m^T F \tag{8}$$

Substituting W_m and (6), (5) into (3), we derive the following objective function:

$$\begin{aligned} \min_{P_m, \alpha_m} \quad & Tr\left(P^T Z P\right) + \mu \sum_{m=1}^{M} (\alpha_m - \theta_m)^2 \\ s.t. \quad & W^T Y W = I, \\ & \sum_{m=1}^{M} \alpha_m = 1, \\ & \sum_{m=1}^{M} \theta_m = 1 \end{aligned} \tag{9}$$

where $P = [P^T_1,\ldots,P^T_M]^T$, $W = [W^T_1,\ldots,W^T_M]^T$, Z consists of several sub-matrices, and its sub-matrix in i-th row and j-th column is:

$$Z_{ij} = \alpha_i \alpha_j \sum_{m=1}^{M} \theta_m G_{ij}^m \tag{10}$$

where:

$$\begin{aligned} G_{ij}^m = & V_i^T D_m V_j - V_i^T V_m V_m^T V_j \\ & - V_i^T X_m \left(X_m^T X_m + \beta_m I_m\right)^{-1} X_m^T V_j \end{aligned} \tag{11}$$

Y also consits of sub-matrices, and its sub-matrix is computed by:

$$Y_{ij} = \alpha_i \alpha_j V_i^T V_j \tag{12}$$

Suppose $V_m = [(V^m_1)^T,\ldots,(V^m_N)^T]^T$, we can obtain an efficient computation of D_m, for each diagonal element in D_m we compute it by:

$$D_i^m = V_i^m \left(\sum_{j=1}^{N} V_j^m\right)^T \tag{13}$$

The total computation time for D_m is $O(N E'^2)$.

The objective function (9) is nonconvex, but it is convex respect to each parameter, thus we propose an alternating process for optimization.

1. Optimizing P_m. All α_m and θ_m are fixed. Discarding the irrelevant terms in (9), we have the following problem:

$$\min_{P_m,\alpha_m} \; Tr\left(P^T Z P\right) \quad s.t. \; W^T Y W = I \tag{14}$$

(14) can be solved by eigenvalue decomposition. P is obtained with the K eigenvectors, which correspond to the K smallest eigenvalues of the generalized eigenvalue problem $Z\,P = \lambda\, Y\,P$.

2. Optimizing α_m. All P_m are fixed, and we adopt the coordinate descent to optimize $\alpha_m \,|^{M}_{m=1}$. In each iteration, we select two elements to update and fix others. Suppose α_i and α_j are selected, since $\sum^{M}_{m=1} \alpha_m = 1$, $\alpha_i + \alpha_j$ will not change in this iteration. Therefore, we obtain the following solution for updating α_i and α_j:

$$\begin{cases} \alpha_i^* = \frac{(2F_{jj} - F_{ij} - F_{ji} + \mu)(\alpha_i + \alpha_j) + \mu(\theta_j - \theta_i)}{2(F_{ii} + F_{jj} - F_{ij} - F_{ji}) + 2\mu} \\ \alpha_j^* = \alpha_i + \alpha_j - \alpha_i^* \end{cases} \tag{15}$$

where:

$$F_{ij} = \sum_{m=1}^{M} \theta_m Tr\left(P_i^T G_{ij} P_j\right) \tag{16}$$

The obtained α^*_i and α^*_j may violate the constraint $\alpha_m > 0$. Thus if $\alpha^*_i < 0$, we set $\alpha^*_i = 0$, and if $\alpha^*_j < 0$, we set $\alpha^*_j = 0$.

3. Optimizing θ_m. All P_m are fixed, and we also adopt the coordinate descent to optimize $\theta_m|^{M}_{m=1}$. In each iteration, the updating will follow the rule of:

$$\begin{cases} \theta_i^* = 0, \theta_j^* = \theta_i + \theta_j, \text{if } \mu\left(\theta_i + \theta_j\right) + S_j - S_i \le 0 \\ \theta_j^* = 0, \theta_i^* = \theta_i + \theta_j, \text{if } \mu\left(\theta_i + \theta_j\right) + S_i - S_j \le 0 \\ \theta_i^* = \frac{\mu(\theta_i + \theta_j) + S_j - S_i}{2\mu}, \theta_j^* = \theta_i + \theta_j - \theta_i^*, \text{else} \end{cases} \tag{17}$$

where:

$$S_m = \sum_{i,j=1}^{M} \alpha_i \alpha_j Tr\left(P_i^T G_{ij}^m P_j\right) \tag{18}$$

In each step, we first optimize α_i and α_j, then we optimize θ_i and θ_j, we iterate this step for all α_m and θ_m.

The whole alternating optimization process is illustrated in Algorithm 1. Note that in order to obtain a relatively concise training process, we do not consider the constrain $W^T Y W = I$ for the optimizing of α_m, but after the optimization of P_m, this constrain is guaranteed. In each updating, the objective function is not increased, thus the constrains and convergence are guaranteed in this algorithm.

Algorithm 1 Training process of MGCMH

Require:

$X_m|_{m=1}^M$

Ensure:

$W_m|_{m=1}^M$, $\alpha_m|_{m=1}^M$, $\theta_m|_{m=1}^M$, H

Randomly choose E training examples, compute $A_{EE}^m|_{m=1}^M$ and $A_{NE}^m|_{m=1}^M$;
For each modality m, compute the E'_m largest eigenvalue of A_{EE}^m, obtain the diagonal matrix Λ_m of eigenvalues, and their corresponding eigenvectors U_m;
Compute $V_m|_{m=1}^M$ by $V_m = A_{NE}^m U_m \Lambda_m^{-1/2}$;
Compute D_m according to (13);
Compute $G_{ij}|_{i,j=1}^M$ according to (11);
Compute Z according to (10);
Compute Y according to (12);
Initialize $\alpha_m|_{m=1}^M$ and $\theta_m|_{m=1}^M$ to $\frac{1}{M}$;
while $t < T$ **do**
    Update $P_m|_{m=1}^M$ by solving the eigenvalue problem of (14);
    if Converge **then**
        Stop iteration;
    end if
    while $\alpha_m|_{m=1}^M$ and $\theta_m|_{m=1}^M$ are not all updated **do**
        Choose i-th and j-th modality;
        Compute $F_{ij}|_{i,j=1}^M$ according to (16);
        Update α_i and α_j according to (15);
        Compute $S_m|_{m=1}^M$ according to (18);
        Update θ_i and θ_j according to (17);
    end while
    t=t+1;
end while
Compute H according to (6);
Compute $W_m|_{m=1}^M$ according to (8);

3.4 Discussion

We can easily find that the complexity of MGCMH is less than $O(N^2)$. In the training process, since the feature dimensions and modality number and are fixed, we only consider N. For the step 1-8 of Algorithm 1, the computing time is about $O(N E^2)$. For the iteration steps, the time complexity is about $O(T E'^3)$, where $E' < E$. Generally the maximum iteration number T is much less than N, and value of sample size E is set to be similar to feature dimensions. Thus we can ignore E, T and E'. As a result, the overall time complexity of training process is linear to the size of training set.

Given a new document, if it contains only one modality, we can use (2) to compute its hash score f. If this document is multi-modal, we first compute the hash scores of each modality f_m, then we combine all modalities to obtain the final scores by $f = \sum_{m=1}^M \alpha_m f_m$. After we obtain the hash scores, we compute the hash codes by $f = \mathrm{sgn}(f - 1/N \sum_{n=1}^N f_n)$, where f_n is the hash scores of the training document.

4 Experiments

4.1 Datasets

In this paper, two real world multi-modal datasets: Wikipedia [24] and NUS-WIDE [5] are used for evaluation. They both consist of image-text pairs and are fully labeled. The statistics of them are summarized in Table 2.

- **Wikipedia** is first used in [24], it is assembled from the "Wikipedia feature articles". It contains 2,866 image-text pairs which are labeled with 10 semantic labels. These labels are used as the ground truth, documents share same concepts with the query are regarded as relevant. Each image is represented by 128-D SIFT histograms, and each text is represented by 10-D LDA histograms. We use 693 pairs as as the query set and the remaining 2,173 pairs as database, which directly correspond to the test/training set in [24].
- **NUS-WIDE** contains 269,648 image-tag pairs downloaded from Flickr, as well as ground-truth for 81 labels that can be used for evaluation. In our experiments, we preserve 10 largest concepts and the corresponding 186,643 pairs. Image features are 500-D SIFT histograms, and text features are 1,000-D vectors represented by the presence of 1,000 tags. We use 1 % of all pairs as query set and the rest as database.

Table 2 The statistics Wikipedia and NUS-WIDE

Datasets	**Wikipedia**	**NUS-WIDE**
Total Size	2,173	186,643
Image Queries	693	1,866
Text Queries	693	1,866
Dimension of image feature	128	500
Dimension of text feature	10	1,000
Database Size	2,173	184,777

It should be noted that in our experiments, the labels in both datasets are only used for evaluation, but not used for training.

4.2 Evaluation metrics

We adopt non-interpolated mean average precision (MAP) to measure the performance. Given a query, AP score is the average of precision obtained for the set of top-R results [8], it is defined as:

$$AP = \frac{1}{p}\sum_{i=1}^{R} pre(i)rel(i) \tag{19}$$

where p is the number of relevant documents in the retrieved set, $pre(i)$ is the precision of top i retrieved documents. $rel(i)$=1 if the i-th retrieved documents is relevant to query, otherwise $rel(i)$=0. The MAP score is the mean of AP scores from all the queries. in this paper we set R = 50. Besides MAP, we also use precision-recall curves to measure the performance.

4.3 Compared methods and implementation details

We compared our method to several representative unsupervised cross-modal hashing methods, including Cross-View Hashing (CVH) [15], Composite Hashing with Multiple Information Sources (CHMIS) [46], Inter-Media Hashing (IMH) [27] and Collective Matrix Factorization Hashing (CMFH) [8]. Since CHMIS and IMH have to compute the graph matrix and compute the eigenvalue of the $N \times N$ matrix, their time complexities are higher than $O(N^2)$. The time complexity of CMFH is also larger than $O(N^2)$. Therefore, our hashing method is more efficient than the compared methods except CVH. Since IMH and CHMIS have to construct full graph matrix for learning, on NUS-WIDE we select 10,000 documents for the training of all methods.

In the implementation of our method, the sample size E is set to 200 on Wikipedia and 500 on NUS-WIDE. The embedding dimension of image E'_1 is set to 100 and 300 on Wikipedia and NUS-WIDE respectively. The embedding dimension of text E'_2 is set to 10 and 30 on two datasets respectively. Since our method is not sensitive to parameters, we set all β_m and γ to 1. In order to make two types of modality weights consistent, we set μ to 10.

4.4 Experimental results

At first we show the modality weights which are automatically learned in our training process. Table 3 shows the image weights α_1, θ_1 and text weights α_2, θ_2 respectively. We can find that that text weights are significantly larger than image weights, which illustrates that text is more important than image in constructing multi-modal graph and generating hash codes. Generally, text contains more semantic information than image content, and previous study also shows that text is more important for cross-modal retrieval [37].

Table 3 Modality weights on two datasets

Wiki	**Code Length**			
	16	**32**	**64**	**128**
α_1	0.0457	0.0232	0.0210	0.0097
α_2	0.9543	0.9768	0.9790	0.9903
θ_1	0.2397	0	0	0
θ_2	0.7603	1	1	1
NUS	16	32	64	128
α_1	0.2146	0.1968	0.1818	0.1663
α_2	0.7854	0.8032	0.8182	0.8337
θ_1	0	0	0	0
θ_2	1	1	1	1

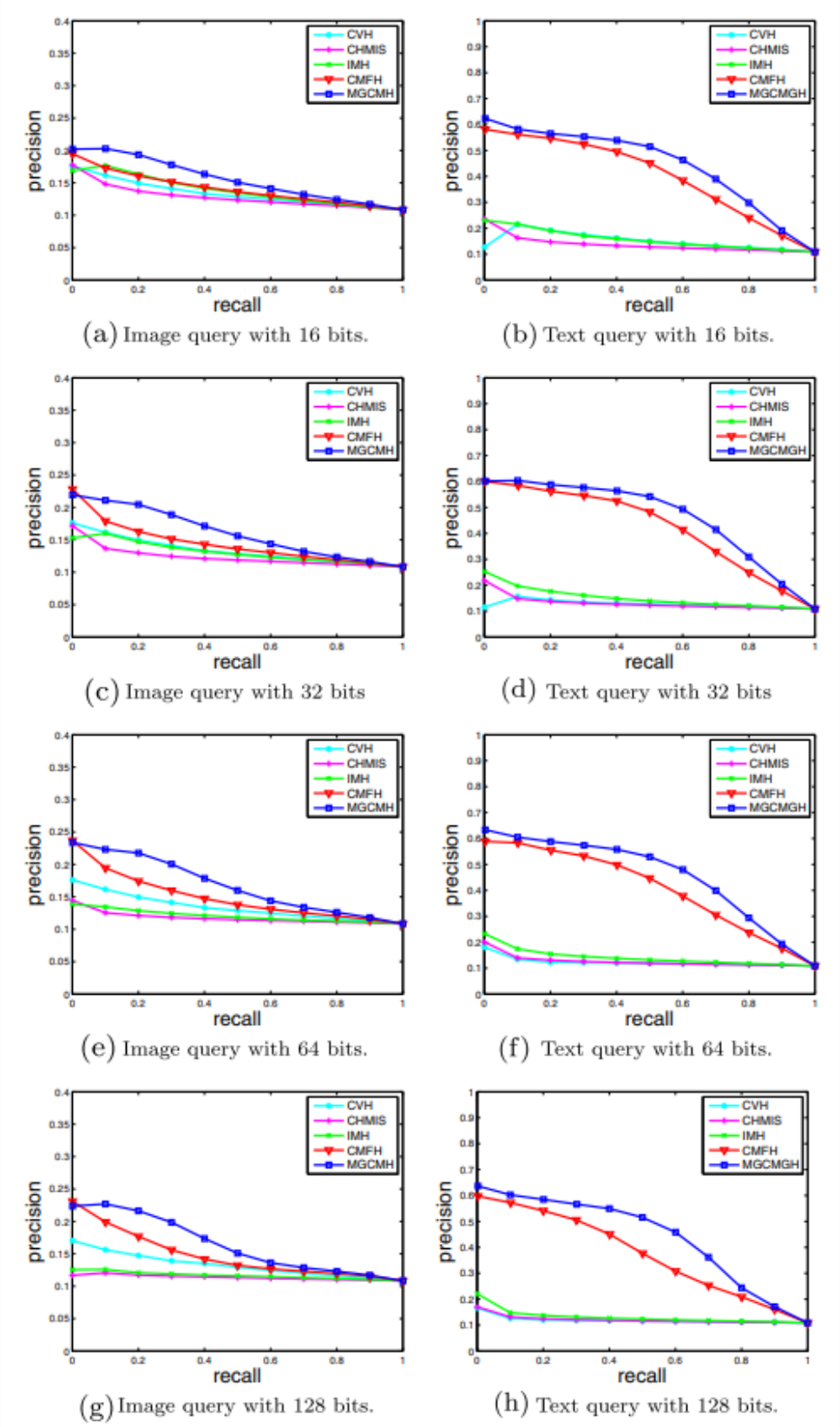
In our experiments, two types of cross-modal search task are considered. The first is image query, where images in the query set are used to search texts in the database. The other is text query, where texts are used to search images.

Table 4 shows the MAP scores of all compared cross-modal hashing methods with varying code length on Wikipedia dataset. We can find that MGCMH and CMFH obtain much higher MAP scores than other methods, this phenomenon is especially significant in text query. The reason is that both MGCMH and CMFH combine multiple modalities in the database and use unified hash codes to represent both images and texts, while other methods only consider single modality in the database. Besides, MGCMH performs better than CMFH. One reason is that the graph framework in MGCMH can handle complex structure of multi-modal data. As a result, hash codes represented by MGCMH can better correlate different modalities than CMFH. The other reason is that modality weights in MGCMH can reflect different contributions of images and texts in the learning process. On the contrary, CMFH does not consider the modality weights, thus it preserves less semantic correlation than MGCMH. In addition, we also observe a desirable characteristic of MGCMH, that it has promising performance with small code length. When using 16 bits codes, the MAP scores of MGCMH are much better than IMH, CHMIS, and CVH with any code lengths, and they are very close to the best scores of CMFH. Fig. 2 shows the PR curves of all compared methods with code length 16, 32, 64 and 128 bits. We can easily find that the results in Fig. 2 is consistent with Table 4, which further demonstrates the superiority of MGCMH in comparison with other methods.

Table 4 MAP scores of hashing methods on Wikipedia

Wiki	**Method**	**Code Length**			
		16	**32**	**64**	**128**
Image query	CVH	0.2132	0.2132	0.2132	0.2051
	CHMIS	0.2123	0.2055	0.1962	0.1818
	IMH	0.2225	0.2004	0.1758	0.1610
	CMFH	0.2490	0.2495	0.2633	0.2670
	MGCMH	0.2540	0.2588	0.2755	0.2860
Text query	CVH	0.2919	0.2471	0.2275	0.1931
	CHMIS	0.2559	0.2485	0.2339	0.2120
	IMH	0.2975	0.3030	0.2645	0.2475
	CMFH	0.6034	0.6231	0.6246	0.6296
	MGCMH	0.6278	0.6439	0.6471	0.6497

Fig. 2 The precision-recall curves on Wikipedia with different code lengths



(a) Image query with 16 bits. (b) Text query with 16 bits.

(c) Image query with 32 bits (d) Text query with 32 bits

(e) Image query with 64 bits. (f) Text query with 64 bits.

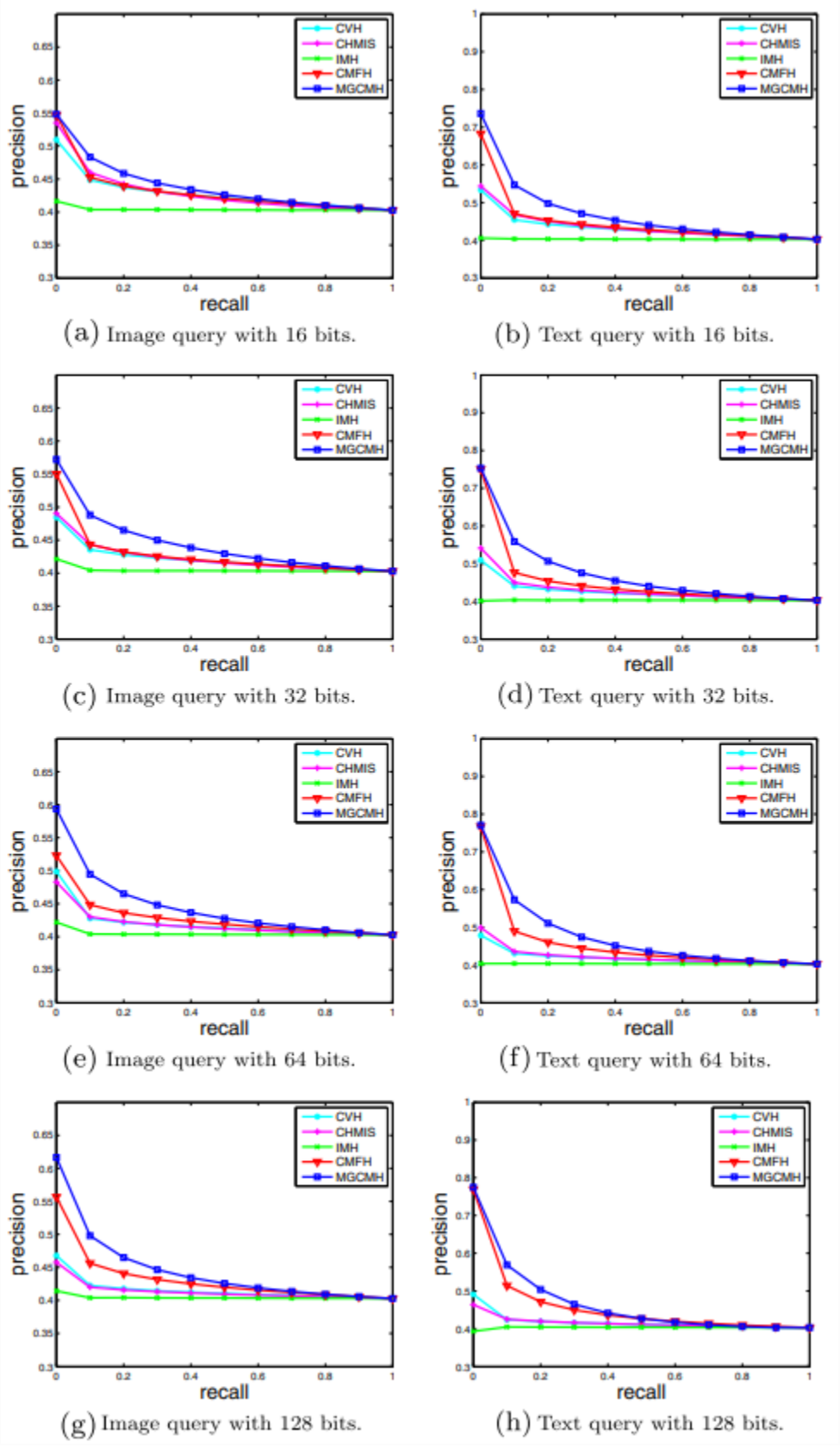
(g) Image query with 128 bits. (h) Text query with 128 bits.

Table 5 shows the MAP scores of all hashing methods with varying code length on NUS-WIDE. We can easily observe similar results on NUS-WIDE, that MGCMH significantly outperforms other methods. Since NUS-WIDE has much more data than Wikipedia, its retrieval tasks are more challenging and more close to the real-world. However, on NUS-WIDE, the increase of MAP scores obtained by MGCMH is more significant than Wikipedia. This phenomenon demonstrates the robustness of MGCMH, that its performance can be guaranteed for large-scale data. Fig. 3 shows the precision-recall curves of all hashing methods on NUS-WIDE with different code lengths. We can find the results are consistent with MAP scores in Table 5. The curves of MGCMH are always higher than curves of other methods.

Table 5 MAP scores of hashing methods on NUS-WIDE

NUS	**Method**	**Code Length**			
		16	**32**	**64**	**128**
Image query	CVH	0.4833	0.4635	0.4500	0.4430
	CHMIS	0.4948	0.4799	0.4605	0.4601
	IMH	0.4567	0.4654	0.4607	0.4628
	CMFH	0.5593	0.5725	0.5874	0.5836
	MGCMH	0.6143	0.6094	0.6281	0.6472
Text query	CVH	0.4866	0.4627	0.4508	0.4444
	CHMIS	0.5429	0.5202	0.4884	0.4612
	IMH	0.4506	0.4477	0.4525	0.4532
	CMFH	0.6987	0.7090	0.7250	0.7313
	MGCMH	0.7360	0.7647	0.7784	0.7743

Fig. 3 The precision-recall curves on NUS-WIDE with different code lengths



(a) Image query with 16 bits. (b) Text query with 16 bits.

(c) Image query with 32 bits. (d) Text query with 32 bits.

(e) Image query with 64 bits. (f) Text query with 64 bits.

(g) Image query with 128 bits. (h) Text query with 128 bits.

At last, we also demonstrate the detailed performance comparison on all 10 concepts of NUS-WIDE. For each concept, we choose the query examples which belong to it to form a new query set, and then we evaluate the MAP scores of this concept. Table 6 shows the MAP scores on each of the 10 concepts of NUS-WIDE, where we set the code length as 128 bits. We can find that MGCMH significantly outperforms other compared methods on most concepts, which confirms the robustness of MGCMH. The only exception is the image query of concept 'person', where MGCMH obtains slightly smaller score than the best result 0.5699. We can also find that all methods perform differently on the 10 concepts, the reason is that some concepts are more difficulty to be represented by hash codes. Moreover, on these difficult concepts such as 'plants', the superiority of MGCMH is more significant.

Table 6 MAP scores on each concept of NUS-WIDE with 128 bits code length

Image query	Animal	Buildings	Clouds	Grass	Ocean	Person	Plants	Sky	Water	Window
CVH	0.4582	0.4271	0.5057	0.4050	0.3079	0.4761	0.3457	0.4901	0.4045	0.4069
CHMIS	0.4665	0.4260	0.5067	0.4179	0.3131	0.5699	0.3419	0.5289	0.4270	0.4247
IMH	0.4279	0.4111	0.5082	0.4059	0.3431	0.5387	0.3588	0.5381	0.4461	0.4050
CMFH	0.5308	0.4572	0.6349	0.4778	0.4080	0.5538	0.3740	0.6211	0.5059	0.4499
MGCMH	0.5742	0.5834	0.7549	0.5271	0.4655	0.5550	0.4688	0.7296	0.5980	0.5398
Text Query										
CVH	0.4515	0.4060	0.5038	0.4158	0.3449	0.5350	0.3213	0.4903	0.4310	0.4012
CHMIS	0.4767	0.4237	0.5114	0.4278	0.3611	0.5748	0.3537	0.5324	0.4700	0.4359
IMH	0.4595	0.4280	0.5061	0.4296	0.3402	0.5644	0.3429	0.5481	0.4677	0.4194
CMFH	0.8086	0.5933	0.6997	0.6979	0.6406	0.7325	0.6451	0.7105	0.7572	0.5828
MGCMH	0.8394	0.6638	0.7887	0.7583	0.7348	0.7997	0.7208	0.7906	0.8473	0.6769

5 Conclusions and future work

In this paper, we propose an unsupervised hashing method: Multi-graph Cross-modal Hashing (MGCMH) for large-scale multimedia search. MGCMH integrates multi-graph learning and hash function learning into a joint framework. Nyström method approximation is used for the efficient construction of graphs. Appropriate weights are assigned in both multi-graph and hash code generation. Then an alternating training process is proposed for the optimization of MGCMH to simultaneously learn hash codes and functions. Finally, the experimental results on two multi-modal datasets demonstrate the effectiveness of MGCMH compared with other representative unsupervised cross-modal hashing methods.

We have confirmed that graph approach is effective in cross-modal hashing. In order to obtain better hashing performance, we can improve our method by introducing multi-modal graph with more complex structure, such as multi-modal hypergraph [52]. However, with the increase of complexity in graph structure, designing an efficient graph construction approach whose time complexity may be linear to the training size, will become more difficult. Therefore, it is challenging to design a complex but efficient multi-modal graph for hashing.

References

1. Andoni A, Indyk P (2006) Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun ACM 51(1):459–468.
2. Bronstein MM, Bronstein AM, Michel F, Paragios N (2010) Data fusion through cross-modality metric learning using similarity-sensitive hashing. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition IEEE, pp 3594–3601.
3. Chang CC, Lin CJ (2011) Libsvm: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27CrossRef.
4. Cheng J, Leng C, Li P, Wang M, Lu H (2014) Semi-supervised multi-graph hashing for scalable similarity search. Comput Vis Image Underst 124:12–21CrossRef.
5. Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) 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.
6. Costa Pereira J, Coviello E, Doyle G, Rasiwasia N, Lanckriet GR, Levy R, Vasconcelos N (2014) On the role of correlation and abstraction in cross-modal multimedia retrieval. IEEE Trans Pattern Anal Mach Intell 36(3):521–535.
7. Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV, Prague, vol 1, pp 1–2.
8. Ding G, Guo Y, Zhou J (2014) Collective matrix factorization hashing for multimodal data. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp 2083– 2090.
9. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) Liblinear: a library for large linear classification. J Mach Learn Res 9:1871–1874zbMATH.
10. Feng F, Wang X, Li R (2014) Cross-modal retrieval with correspondence autoencoder. In: Proceedings of the ACM International Conference on Multimedia, pp 7–16.
11. Gao L, Song J, Nie F, Yan Y, Sebe N, Tao Shen H (2015) Optimal graph learning with partial tags and multiple features for image and video annotation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4371–4379.
12. Guillaumin M, Verbeek J, Schmid C (2010) Multimodal semi-supervised learning for image classification. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition IEEE, pp 902–909.
13. He K, Wen F, Sun J (2013) K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp 2938–2945.
14. Hotelling H (1936) Relations between two sets of variates. Biometrika:321–377.
15. Kumar S, Udupa R (2011) Learning hash functions for cross-view similarity search. In: Proceedings of the international joint conference on artificial intelligence, vol 22, p 1360.
16. Li D, Dimitrova N, Li M, Sethi IK (2003) Multimedia content processing through cross-modal association. In: Proceedings of the 11th ACM international conference on Multimedia, ACM, pp 604–611.
17. Liu W, Mu C, Kumar S, Chang SF (2014) Discrete graph hashing. In: Proceedings of NIPS, pp 3419–3427.
18. Liu W, Tao D (2013) Multiview hessian regularization for image annotation. IEEE Trans Image Process 22(7):2676–2687.
19. Liu W, Wang J, Kumar S, Chang SF (2011) Hashing with graphs. In: Proceedings of the 28th international conference on machine learning, pp 1–8.
20. Luo Y, Liu T, Tao D, Xu C (2015) Multiview matrix completion for multilabel image classification. IEEE Trans Image Process 24(8):2355–2368.
21. Ma Z, Yang Y, Sebe N, Hauptmann AG (2014) Multiple features but few labels? A symbiotic solution exemplified for video analysis. In: Proceedings of the ACM International Conference on Multimedia, ACM, pp 77–86.
22. Ni B, Moulin P, Yan S (2015) Order preserving sparse coding. IEEE Trans Pattern Anal Mach Intell 37:1615–1628.

23. Perronnin F, Snchez J, Mensink T (2010) Improving the fisher kernel for large-scale image classification. In: Proceedings of the 11th European Conference on Computer Vision, pp 143–156.
24. Rasiwasia N, Costa Pereira J, Coviello E, Doyle G, Lanckriet GR, Levy R, Vasconcelos N (2010) A new approach to cross-modal multimedia retrieval. In: Proceedings of the 18th ACM international conference on Multimedia, ACM, pp 251–260.
25. Saberian MJ, Vasconcelos N (2011) Multiclass boosting: Theory and algorithms. In: Proceedings of NIPS, pp 2124–2132.
26. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge: Cambridge University Press.
27. Song J, Gao L, Yan Y, Zhang D, Sebe N (2015) Supervised hashing with pseudo labels for scalable multimedia retrieval. In: Proceedings of the 23rd ACM Conference on Multimedia, ACM, pp 827– 830.
28. Song J, Yang Y, Yang Y, Huang Z, Shen HT (2013) Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp 785–796.
29. Sonnenburg S, Rätsch G, Schäfer C, Schölkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565.
30. Wang J, Kumar S, Chang SF (2010) Semi-supervised hashing for scalable image retrieval. In: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition IEEE, pp 3424– 3431.
31. Wang J, Kumar S, Chang SF (2012) Semi-supervised hashing for large-scale search. IEEE Trans Pattern Anal Mach Intell 34(12):2393–2406.
32. Wang K, He R, Wang W, Wang L, Tan T (2013) Learning coupled feature spaces for cross-modal matching. In: Proceedings of 2013 IEEE International Conference on Computer Vision IEEE, pp 2088–2095.
33. Wang Q, Si L, Zhang Z, Zhang N (2014) Active hashing with joint data example and tag selection. In: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM, pp 405–414.
34. Wang W, Yang X, Ooi BC, Zhang D, Zhuang Y (2015) Effective deep learning-based multi-modal retrieval. VLDB J:1–23.
35. Weiss Y, Torralba A, Fergus R (2009) Spectral hashing. In: Proceedings of NIPS, pp 1753–1760.
36. Xie L, Pan P, Lu Y, Wang S (2014) A cross-modal multi-task learning framework for image annotation. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, pp 431–440.
37. Xie L, Pan P, Lu Y (2015) Analyzing semantic correlation for cross-modal retrieval. Multimedia Systems 21(6):525–539.
38. Xie L, Zhu L, Pan P, Lu Y (2015) Cross-modal self-taught hashing for large-scale image retrieval. Signal Processing.
39. Yan Y, Ricci E, Subramanian R, Liu G, Lanz O, Sebe N (2015) A multi-task learning framework for head pose estimation under target motion.
40. Yan Y, Ricci E, Subramanian R, Liu G, Sebe N (2014) Multitask linear discriminant analysis for view invariant action recognition. IEEE Trans Image Process 23(12):5599–5611.
41. Yan Y, Yang Y, Meng D, Liu G, Tong W, Hauptmann AG, Sebe N (2015) Event oriented dictionary learning for complex event detection. IEEE Trans Image Process 24(6):1867–1878.
42. Yang Y, Zhuang YT, Wu F, Pan YH (2008) Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. IEEE Trans Multimedia 10(3):437–446.
43. Yang Y, Xu D, Nie F, Luo J, Zhuang Y (2009) Ranking with local regression and global alignment for cross media retrieval. In: Proceedings of the 17th ACM international conference on Multimedia, ACM, pp 175–184.
44. Zhang D, Li WJ (2014) Large-scale supervised multimodal hashing with semantic correlation maximization. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp 2177– 2183.
45. Zhang D, Wang J, Cai D, Lu J (2010) Self-taught hashing for fast similarity search. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp 18–25.

46. Zhang D, Wang F, Si L (2011) Composite hashing with multiple information sources. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, pp 225–234.
47. Zhang K, Tsang IW, Kwok JT (2008) Improved Nyström low-rank approximation and error analysis. In: Proceedings of the 25th international conference on Machine learning, ACM, pp 1232–1239.
48. Zhang K, Kwok JT, Parvin B (2009) Prototype vector machine for large scale semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, pp 1233– 1240.
49. Zhang P, Zhang W, Li WJ, Guo M (2014) Supervised hashing with latent factor models. In: Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 173–182.
50. Zhen Y, Yeung DY (2012) A probabilistic model for multimodal hash function learning. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 940–948.
51. Zhen Y, Yeung DY (2013) Active hashing and its application to image and text retrieval. Data Min Knowl Disc 26(2):255–274.
52. Zhu L, Shen J, Jin H, Zheng R, Xie L (2015) Content-based visual landmark search via multimodal hypergraph learning. IEEE Transactions on Cybernetics.
53. Zhu L, Shen J, Xie L (2015) Topic hypergraph hashing for mobile image retrieval. In: Proceedings of the 23rd ACM Conference on Multimedia Conference, ACM, pp 843–846.
54. Zhu X, Huang Z, Shen HT, Zhao X (2013) Linear cross-modal hashing for efficient multimedia search. In: Proceedings of the 21st ACM international conference on Multimedia, ACM, pp 143– 152.