Iterative Multi-View Hashing for Cross Media Indexing

Yao Hu* Zhongming Jin* Hongyi Ren Deng Cai Xiaofei He
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China
{yaoohu, jinzhongming888, erichongyi, dengcai, xiaofeihe}@gmail.com

ABSTRACT

Cross media retrieval engines have gained massive popularity with rapid development of the Internet. Users may perform queries in a corpus consisting of audio, video, and textual information. To make such systems practically possible for large mount of multimedia data, two critical issues must be carefully considered: (a) reduce the storage as much as possible; (b) model the relationship of the heterogeneous media data. Recently academic community have proved that encoding the data into compact binary codes can drastically reduce the storage and computational cost. However, it is still unclear how to integrate multiple information sources properly into the binary code encoding scheme.

In this paper, we study the cross media indexing problem by learning the discriminative hashing functions to map the multi-view datum into a shared hamming space. Not only meaningful within-view similarity is required to be preserved, we also incorporate the between-view correlations into the encoding scheme, where we map the similar points close together and push apart the dissimilar ones. To this end, we propose a novel hashing algorithm called *Iterative Multi-View Hashing* (IMVH) by taking these information into account simultaneously. To solve this joint optimization problem efficiently, we further develop an iterative scheme to deal with it by using a more flexible quantization model. In particular, an optimal alignment is learned to maintain the between-view similarity in the encoding scheme. And the binary codes are obtained by directly solving a series of binary label assignment problems without continuous relaxation to avoid the unnecessary quantization loss. In this way, the proposed algorithm not only greatly improves the retrieval accuracy but also performs strong robustness. An extensive set of experiments clearly demonstrates the superior performance of the proposed method against the state-of-the-art techniques on both multimodal and unimodal retrieval tasks.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

*Equal contribution.

MM'14, November 03 - 07 2014, Orlando, FL, USA

http://dx.doi.org/10.1145/2647868.2654906 .

General Terms

Algorithms

Keywords

Multi-View Hashing; Within-View Similarity; Between-View Correlations.

1. INTRODUCTION

In recent years, with the explosive growth of the available media data, hashing method has attracted many researchers' attention due to its great advantages in reducing both the computational cost and storage. A lot of hashing algorithms have been proposed for various applications, such as image and video retrieval [28, 27, 31, 12], duplicate image detection [3], key point detection [29] and so on.

However, most of the existing hashing methods are mainly designed for single type of data, such as sift or gist feature for image [13] and Tf-idf feature for text [8]. In these methods, the hashing functions are designed to learn similarity preserving binary codes for data representation. However, it is well known that multimedia data with same semantics can exist in more than one view [21, 5, 19]. For example, we can describe one topic by text document, image or audio. In each view, the corresponding type of feature only reveals the partial information. The joint consideration of multiple features in the multi-view space can assist us to better understand the underlying data distribution. To model the mutual correlation across different views, traditional cross media retrieval algorithms firstly project all the original features into a shared semantic correlation space [23, 30] and then perform alignments for pair matching between views [18]. However, it is still a puzzle how to construct an effective indexing based on these algorithms.

To build an efficient indexing for cross media retrieval, a series of multi-view hashing algorithms have been proposed to encode the multimodal data sources from different perspectives [24, 33, 15]. Not only the respective information from each individual view but also the mutual information across different views are required to be simultaneously preserved for designing discriminative multi-view hashing functions. Specifically, just as pointed out in [24, 32], we expect to ensure the consistency between the learned hashing codes and the corresponding hashing functions designed for different information sources, which means that small variations of the data should not result in prominent difference for the final binary codes. Furthermore, it is also necessary to push apart the data points with different concepts, no matter in which view they are available, to make the encoding scheme more discriminative [22]. Such properties are understood as between-view similarity and between-view distinctiveness preservation respectively when multi-view data representations are mapped into the

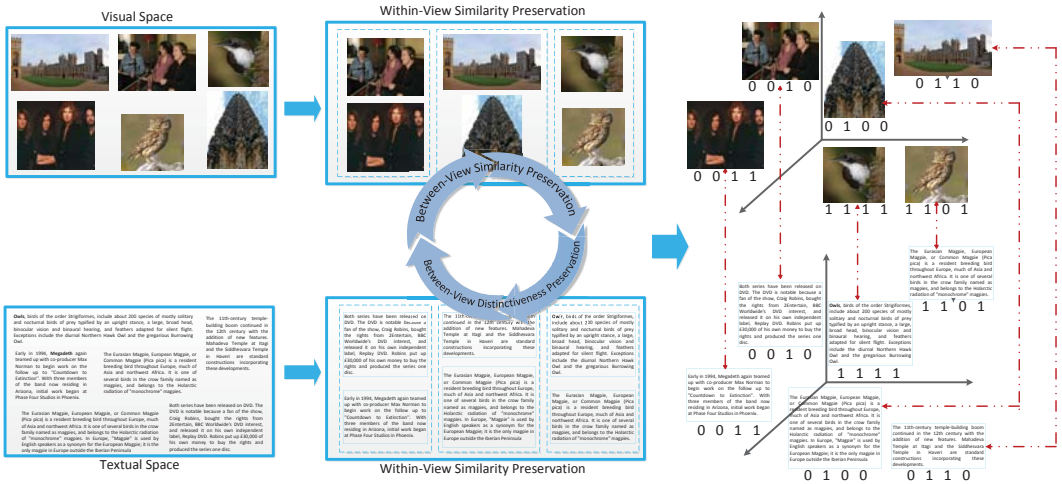


Figure 1: The illustration of our proposed Iterative Multi-View Hashing (IMVH).

produced common Hamming space. In [33], the authors attempt to preserve these two types of information simultaneously in the Euclidean space. However, since the rough incorporation with a weak representative ability for the between-view correlations, it does not take full advantage of the mutual information across different views, which leads a limited performance. Therefore, how to incorporate these two types of information into the multi-view hashing function learning effectively still remains challenging.

To address this problem, in this paper we propose a novel multi-view hashing method called *Iterative Multi-View Hashing* (IMVH) to preserve the within-view similarity and between-view correlations, including between-view similarity and between-view distinctiveness of the original data distribution, into the Hamming space in an iterative scheme. An illustration of the whole flowchart is shown in Figure 1. We begin with learning hashing functions in each view involving the within-view similarity and between-view distinctiveness preservation term, while the between-view similarity is incorporated globally during the optimization process. Specifically, in each iteration, an optimal rotation is learned to align the data points belonging to the same concept in different views to globally preserve the between-view similarity. Furthermore, inspired by the recent proposed idea regarding the binary codes as auxiliary variables for a more flexible model, we update the binary codes by solving a series of binary label assignment problems to reduce the unnecessary cost caused by the rough quantization with the sign function $\mathrm{sgn}(\cdot)$. In this way, the obtained multi-view hashing functions can be guaranteed with enough consistency and discriminative power. Experimental results on several benchmarks confirm that our proposed IMVH greatly improves accuracy with strong robustness on cross-modal retrieval tasks and unimodal retrieval tasks compared with the state-of-the-art multimodal and unimodal hashing approaches respectively.

For the purpose of explaining our basic idea clearly, in the following we firstly focus on the dual-view case, e.g., image feature space $\mathcal{I}$ and textual feature space $\mathcal{T}$.

Notations: We assume $*$ is a placeholder for space $\mathcal{I}$ or $\mathcal{T}$ and denote $\mathbf{X}_* \in \mathbb{R}^{D_* \times N}$ to be the centralized data matrix whose column $\mathbf{X}_*^i$ represents to the i-th sample from either space. D_* is the dimension of the corresponding space. We further define K bits hashing code of sample x as $h_*(x) = [h_*^1(x), \cdots, h_*^K(x)] \in \{1, -1\}^K$ and $h_*^k(x) = \mathrm{sgn}((w_*^k)^T x)$. For simplicity, we denote the projection matrix $\mathbf{W}_* = [w_*^1, w_*^2, \ldots, w_*^K] \in \mathbb{R}^{D_* \times K}$ and the binary code matrix $\mathbf{B}_*$ can be computed as $\mathbf{B}_* = \mathrm{sgn}(\mathbf{X}_*^T \mathbf{W}_*) \in \mathbb{R}^{N \times K}$ [1]. Similarly the vector $\mathbf{B}_*^k$ represents the k-th column of $\mathbf{B}_*$ (the k-th bits of all samples). To obtain the semantic information, we also assume that each pair $(\mathbf{X}_\mathcal{I}^i, \mathbf{X}_\mathcal{T}^j)$ has a label $s_{ij} = 1$ if they have the same concept and $s_{ij} = 0$ otherwise. For clear description, we also assume each pair $(\mathbf{X}_\mathcal{I}^i, \mathbf{X}_\mathcal{T}^i)$ exists in the same concept, *i.e.*, $s_{ii} = 1$.

2. BACKGROUND

Most of the previously proposed hashing methods for single-view data can be decomposed into two steps: (1) Firstly, project all the original data points into a low dimensional space; (2) obtain the binary codes by quantizing the new data representations in the embedding space. In unsupervised case, a series of methods have been proposed based on the different embedding methods, such as random projection [7], spectral decomposition [28] and other learning schemes [25, 14, 20, 11, 10]. Furthermore, when the semantic information is available, the projection directions are constructed to make the binary codes fit the supervision as much as possible [17].

Recently several hashing algorithms for multi-view data have also been studied. Along the lines of single-view hashing, these hashing algorithms also can be decomposed into the similar two steps imposing a couple of constraints. In [9], Canonical Correlation Analysis (CCA) is adopted to extract a common latent space from two views by solving

$$\begin{aligned} \min_{w_\mathcal{I}, w_\mathcal{T}} \quad & w_\mathcal{I}^T \mathbf{X}_\mathcal{I} \mathbf{X}_\mathcal{T}^T w_\mathcal{T} \\ s.t. \quad & w_*^T \mathbf{X}_* \mathbf{X}_*^T w_* = 1, * = \mathcal{I} \text{ or } \mathcal{T}. \end{aligned} \tag{1}$$

Similar latent semantic space such as Multimedia Correlation Space [30], Correlation Semantic Space [23] based on CCA and Parallel Field Embedding Space [18] based on manifold alignment are also introduced from different perspective. Then a joint model can be constructed here after all the data in different views are projected into such intermediate space. An intuitive way to construct a cross media indexing is to directly quantize the results from these

[1]The corresponding hash bit can be computed as $(1 + h_*(x))/2$ simply .

methods. However, it would leads an unbalance encoding scheme. Furthermore, such methods only extract partial information of the correlations and most of the critical information of the data distribution is ignored, which limits the performance of the learned hashing functions. On the other hand, Rastegari *et al.*[24] propose to find the most discriminative hashing functions by enforcing the maximum margin constraints and a block coordinate descent based iterative scheme is applied for the optimization. To preserve the with-in view and between-view similarity, Kumar *et al.*[15] formulate the multi-view hashing problem as an extension of traditional spectral hashing, while Bronstein *et al.*[2] learn the projection directions from the perspective of boosting. Except for the similarity preservation, the between-view distinctiveness has also been pointed out to be critical for the discriminative power of the learned hashing functions [22]. This insight enables the joint preservation of the between-view similarity and distinctiveness in [33], where the authors simultaneously consider these two regularization terms in the optimization procedure. However, this method suffers from the unsatisfactory performance which may be caused by its improper incorporation of between-view correlations.

3. OUR APPROACH

For multi-view hashing problem, the main challenge is how to incorporate the within-view information and between-view correlations into an unified framework properly. To obtain powerful discriminative hashing functions, we expect to force the dissimilar pairs to be far away in the produced Hamming space. Meanwhile, the global similarity across all the views should also be preserved for consistent encoding scheme. In our approach, we explore to incorporate these different aspects of the original data distribution in a global framework. For better description, in this section, we firstly show how to formulate the within-view similarity and between-view correlations. Then an iterative optimization strategy will be provided for optimization later.

3.1 Within-View Similarity Preservation

Within-view similarity preservation is designed to maintain the neighborhood relationships among the data points in each individual view after being mapped into the produced Hamming space. In traditional single view hashing methods, Spectral Hashing [28] preserves the within-view similarity by minimizing the weighted Hamming distance between codewords correlates with the similarity. While for multi-view hashing methods [32], to deal with the multiple information sources, the similarity quantity for each individual view is measured with the same way as in Spectral Hashing. And the total cost is the summation across all the data modalities.

Motivated by the recent progress in Supervised Hashing methods [27, 17], by defining an N-by-N affinity matrix $\mathbf{S}_*$, we expect the similarity between training data points represented by each bit can approximate $\mathbf{S}_*$ as much as possible. And the within-view similarity for each individual view can be measured as the summation of the costs for all the bits. Then the total within-view similarity for all the views can be preserved by minimizing the following term:

$$\frac{1}{2}\sum_{*}\sum_{k=1}^{K}\|\mathbf{B}_*^k(\mathbf{B}_*^k)^T-\mathbf{S}_*\|_F^2=-\sum_{*}Tr(\mathbf{B}_*^T\mathbf{S}_*\mathbf{B}_*)+const.$$

Notice that there are many choices for the affinity matrix $\mathbf{S}_*$. Intuitively, we adopt the same way as in [27]. We define $\mathbf{S}_*^{ij}=1$ if the pair $(\mathbf{X}_*^i,\mathbf{X}_*^j)$ are denoted as similar, $\mathbf{S}_*^{ij}=-1$ if dissimilar and $\mathbf{S}_*^{ij}=0$ if unknown. By minimizing (2), it incurs a heavy penalty if two similar examples are mapped far away in the Hamming space and then the similarity between different examples can be preserved in the learned hashing codes.

3.2 Between-View Correlations Preservation

For multi-view hashing problems, only preserving the within-view similarity in each view is far from enough. It is critical to incorporate the relationships between different views when learning the hashing functions. Specifically, it is natural to assume that the points with the same concept should be binarized consistently wherever they are available. Therefore, we formulate this between-view similarity preservation term as

$$\Phi(\mathcal{I},\mathcal{T}) \quad = \quad \|\mathbf{B}_{\mathcal{I}}-\mathbf{B}_{\mathcal{T}}\|_F^2.$$

Meanwhile it is also important to note that the points with different concepts should not be mapped closely together, which may seriously lower the discriminative power of the learned hashing function. To meet this requirement, in the following, we propose to penalize the distinctiveness for data points with different concepts across all the views.

Following the similar way in [22], we need to severely punish the dissimilar pairs if they have similar binary codes, *i.e.*, small hamming distance after encoding, which can be realized by minimizing the between-view distinctiveness term. Taking the distinctiveness term $\Omega_{\mathcal{I}}(\mathcal{I},\mathcal{T})$ from $\mathcal{I}\rightarrow\mathcal{T}$ as example, it can be defined as follows:

$$\Omega_{\mathcal{I}}(\mathcal{I},\mathcal{T})=\sum_{i=1}^{N}\ell_i,\quad \ell_i=\frac{\sum_{j=1}^{N}(1-s_{ij})\tau(d_{ij})}{\sum_{j=1}^{N}(1-s_{ij})}, \qquad (2)$$

where $d_{ij}=hamm(h_{\mathcal{I}}(\mathbf{X}_{\mathcal{I}}^i),h_{\mathcal{T}}(\mathbf{X}_{\mathcal{T}}^j))=\frac{1}{2}\sum_{t=1}^{K}|\mathbf{B}_{\mathcal{I}}^{it}-\mathbf{B}_{\mathcal{T}}^{jt}|$ and the distinctiveness term $\Omega_{\mathcal{T}}(\mathcal{I},\mathcal{T})$ from $\mathcal{T}\rightarrow\mathcal{I}$ can also be defined in a similar way. $\tau(d)$ is called the smoothly clipped inverted squared deviation (SCISD) function [22], which is defined as

$$\tau(d)=\begin{cases}\frac{-d^2+ab^2}{2} & \text{if } |d|\le b,\\ \frac{d^2-2ab|d|+a^2b^2}{2(a-1)} & \text{if } b\le|d|\le ab,\\ 0 & \text{if } ab\le|d|,\end{cases} \qquad (3)$$

where a and b are two user-specified parameters.

It is important to note the fact that the between-view distinctiveness preservation term $\Omega_*(\mathcal{I},\mathcal{T})$ in (2) is quite different from the regularization term used in [33], where the distinctiveness is preserved for learning the hashing function based on the Euclidean distance between the projected pairs. While in (2), we utilize Hamming distance instead of the Euclidean distance to precisely measure the difference of the projected pairs. In this way, we measure the distinctiveness penalization of $\mathbf{X}_{\mathcal{I}}^i$ as the average distinctiveness of between all the dissimilar pairs involving $\mathbf{X}_{\mathcal{I}}^i$ across all the views. Therefore, minimizing the regularization term $\Omega_*(\mathcal{I},\mathcal{T})$ can better assist us to find the hash functions which enjoy the large-margin property.

3.3 The Objective Function

Furthermore, a lot of research works [28, 17] have revealed that the independence between hash functions assists the balance of the learned hashing codes in each feature space. Therefore, we also add a regularization $\sum_*\|\mathbf{B}_*^T\mathbf{B}_*-N\mathbf{I}_K\|_F^2$, where $\mathbf{I}_K\in\mathbb{R}^K$ is an identity matrix. After some simple algebraic operations, it is easy to observe that $\sum_*\|\mathbf{B}_*^T\mathbf{B}_*-N\mathbf{I}_K\|_F^2=\sum_*\Big(\|\mathbf{B}_*^T\mathbf{B}_*\|_F^2-2N\|\mathbf{B}_*\|_F^2+N^2\|\mathbf{I}_K\|_F^2\Big)=\sum_*\|\mathbf{B}_*^T\mathbf{B}_*\|_F^2+const.$ Therefore, we can minimize $\sum_*\|\mathbf{B}_*^T\mathbf{B}_*\|_F^2$ instead to force the independence between the learned hashing functions.

Recall that a discriminative hashing functions should ensure the dissimilar data across different views are forced to be pushed apart and similar pairs are binarized consistently. Therefore, the corresponding regularization terms can be incorporated into the objective function as constraints. Furthermore, to obtain better generalization ability, it is also suggested by the recent research advances [16, 24] to maximize margin between positive and negative samples for each hashing function. Overall, we can formulate the objective function of IMVH as

$$\begin{aligned}\mathcal{O} = \Phi(\mathcal{I},\mathcal{T}) + \sum_{*}\Big(\mathcal{L}(\mathbf{B}_*, h_*(\mathbf{X}_*)) + \lambda_*\|\mathbf{W}_*\|_F^2 \\ + \alpha_*\Omega_*(\mathcal{I},\mathcal{T}) - \beta_* Tr(\mathbf{B}_*^T\mathbf{S}_*\mathbf{B}_*) + \gamma_*\|\mathbf{B}_*^T\mathbf{B}_*\|_F^2\Big),\end{aligned} \tag{4}$$

where the loss term $\mathcal{L}(\mathbf{B}_*, h_*(\mathbf{X}_*))$ can be defined as the summation of the hinge loss of all the K hash bits as $\mathcal{L}(\mathbf{B}_*, h_*(\mathbf{X}_*)) = \sum_{k=1}^{K}\mathcal{L}(\mathbf{B}_*^k, h_*^k(\mathbf{X}_*)) = \sum_{k=1}^{K}\sum_{i=1}^{N}\max(0, 1-\mathbf{B}_*^{ik}h_*^k(\mathbf{X}_*^i))$. And the regularization term $\|W\|_F^2$ is proposed to ensure the max-margin constraint for the learned hyperplanes.

To solve problem (4) efficiently, we propose to solve it by using a simple iterative scheme. Firstly we can fix $\mathbf{W}_\mathcal{T}$ and $\mathbf{B}_\mathcal{T}$ and solve the following subproblem in the image feature space:

$$\begin{aligned}\min_{\mathbf{W}_\mathcal{I},\mathbf{B}_\mathcal{I}} \mathcal{O}_\mathcal{I} = \mathcal{L}(\mathbf{B}_\mathcal{I}, h(\mathbf{X}_\mathcal{I})) + \lambda_\mathcal{I}\|\mathbf{W}_\mathcal{I}\|_F^2 + \Phi(\mathcal{I},\mathcal{T}) \\ + \alpha_\mathcal{I}\Omega_\mathcal{I}(\mathcal{I},\mathcal{T}) - \beta_\mathcal{I} Tr(\mathbf{B}_\mathcal{I}^T\mathbf{S}_\mathcal{I}\mathbf{B}_\mathcal{I}) + \gamma_\mathcal{I}\|\mathbf{B}_\mathcal{I}^T\mathbf{B}_\mathcal{I}\|_F^2.\end{aligned} \tag{5}$$

And then we fix $\mathbf{W}_\mathcal{I}$ and $\mathbf{B}_\mathcal{I}$ and minimize the corresponding subproblem in the textual feature space:

$$\begin{aligned}\min_{\mathbf{W}_\mathcal{T},\mathbf{B}_\mathcal{T}} \mathcal{O}_\mathcal{T} = \mathcal{L}(\mathbf{B}_\mathcal{T}, h(\mathbf{X}_\mathcal{T})) + \lambda_\mathcal{T}\|\mathbf{W}_\mathcal{T}\|_F^2 + \Phi(\mathcal{I},\mathcal{T}) \\ + \alpha_\mathcal{T}\Omega_\mathcal{T}(\mathcal{I},\mathcal{T}) - \beta_\mathcal{T} Tr(\mathbf{B}_\mathcal{T}^T\mathbf{S}_\mathcal{T}\mathbf{B}_\mathcal{T}) + \gamma_\mathcal{T}\|\mathbf{B}_\mathcal{T}^T\mathbf{B}_\mathcal{T}\|_F^2.\end{aligned} \tag{6}$$

By solving the above two subproblems alternatively, we can obtain the satisfactory hashing functions with outperforming discriminative power meanwhile preserving relationships between different views. In spite of the seemingly complex appearance of (5) and (6), they can be solved by a quite efficient way. In the next section, we take the subproblem (5) as an example to show the detailed optimization procedure. And the subproblem (6) also can be solved in the same way.

4. OPTIMIZATION METHOD

Note that the between-view correlations preservation terms $\Phi(\mathcal{I},\mathcal{T})$ and $\Omega_\mathcal{I}(\mathcal{I},\mathcal{T})$ in $\mathcal{O}_\mathcal{I}$ are both non-convex and non-smooth since their respective special configuration. Therefore, the main challenge is how to deal with the joint optimization problem in an unified scheme. Following the similar framework proposed by Lin *et al*. [16], we view the binary code matrix $\mathbf{B}_\mathcal{I}$ as an auxiliary variable for hash function learning and can be deviated from $\text{sgn}(\mathbf{X}_\mathcal{I}^T\mathbf{W}_\mathcal{I})$, which makes the optimization process more flexible. In this section, we propose an iterative scheme to solve the subproblem (5) efficiently , where an optimal rotation matrix is learned to preserve the global between-view similarity for a more discriminative projection matrix $\mathbf{W}_\mathcal{I}$ and then we update $\mathbf{B}_\mathcal{I}$ by solving a series of binary assignment problems.

4.1 Update $\mathbf{W}_\mathcal{I}$ by Optimal Rotation

After we fix $\mathbf{B}_\mathcal{I}, \mathbf{B}_\mathcal{T}$ and $\mathbf{W}_\mathcal{T}$, we firstly update $\mathbf{W}_\mathcal{I}$ by solving

$$\min_{\mathbf{W}_\mathcal{I}} \mathcal{L}(\mathbf{B}_\mathcal{I}, h(\mathbf{X}_\mathcal{I})) + \lambda_\mathcal{I}\|\mathbf{W}_\mathcal{I}\|_F^2. \tag{7}$$

In this way, we actually learn K independent binary SVM classifiers, one for each bit. The learned projection matrix $\mathbf{W}_\mathcal{I}$ can force the the mapped values far away from 0 and then the corresponding hashing functions have a better generalization ability [20]. However, the lack of the critical between-view similarity information makes $\mathbf{W}_\mathcal{I}$ not powerful enough for a consistent encoding scheme.

Motivated by the recent work [9] which proposes to preserve the global similarity by learning an orthogonal rotation matrix to minimize the quantization loss between the hashing codes and original datum, we refine the previously obtained projection matrix $\mathbf{W}_\mathcal{I}$ in a similar way. We expect such rotation can minimize the distortion between different views as much as possible to globally preserve the between-view similarity. Therefore, we relax the between-view similarity preservation term $\Phi(\mathcal{I},\mathcal{T})$ by minimizing $\|\mathbf{X}_\mathcal{I}^T\mathbf{W}_\mathcal{I} - \mathbf{B}_\mathcal{T}\|_F^2$. Then an orthogonal rotation matrix $\mathbf{R}_\mathcal{I}$ is learned for the coarse projected data $\mathbf{V}_\mathcal{I} = \mathbf{X}_\mathcal{I}^T\mathbf{W}_\mathcal{I}$ by minimizing the following optimization problem:

$$\begin{aligned}\min_{\mathbf{R}_\mathcal{I}} \quad & \|\mathbf{V}_\mathcal{I}\mathbf{R}_\mathcal{I} - \mathbf{B}_\mathcal{T}\|_F^2 \\ s.t. \quad & \mathbf{R}_\mathcal{I}^T\mathbf{R}_\mathcal{I} = \mathbf{I}_K, \mathbf{R}_\mathcal{I}\mathbf{R}_\mathcal{I}^T = \mathbf{I}_K.\end{aligned} \tag{8}$$

Note that the binary matrix $\mathbf{B}_\mathcal{T}$ in another feature space is fixed and obtained by the previous iteration. Therefore, minimizing (8) actually is equivalent to finding a rotation to align the data in the current feature space with another, which is a classic Orthogonal Procrustes problem [26]. We firstly compute the SVD of the $K \times K$ matrix $\mathbf{B}_\mathcal{T}^T\mathbf{V}_\mathcal{I}$ and then obtain the optimal solution of (8) as

$$\mathbf{B}_\mathcal{T}^T\mathbf{V}_\mathcal{I} = U\Sigma V^T \quad and \quad \mathbf{R}_\mathcal{I} = VU^T. \tag{9}$$

After learning the optimal rotation matrix, we refine the the projection matrix as

$$\mathbf{W}_\mathcal{I} \longleftarrow \mathbf{W}_\mathcal{I}\mathbf{R}_\mathcal{I}. \tag{10}$$

Obviously the problem (8) actually forces the global consistency between the hash codes of different views. Different from [24] adopting a similar regularization for initialization and preserving consistency during learning SVM classifiers, in our approach the between-view consistency information is considered by learning the optimal rotation. In this way, the between-view similarity can be preserved.

4.2 Update $\mathbf{B}_\mathcal{I}$ by Graph Cut

Different from the traditional hashing techniques [28, 24] binarizing the projected data with sign function $\text{sgn}(\cdot)$ to obtain $\mathbf{B}_\mathcal{I}$, some researchers propose to update this binary matrix by solving binary assignment problems indirectly [16] or directly [6], which reduces the unnecessary loss caused by directly binarization and greatly improves the performance . In our approach, we also take the similar way to deal with this binary assignment procedure to update $\mathbf{B}_\mathcal{I}$. With the fixed $\mathbf{W}_\mathcal{I}$, the subproblem (5) degrades to be

$$\begin{aligned}\min_{\mathbf{B}_\mathcal{I}} \quad & \mathcal{L}(\mathbf{B}_\mathcal{I}, h(\mathbf{X}_\mathcal{I})) + \alpha_\mathcal{I}\Omega_\mathcal{I}(\mathcal{I},\mathcal{T}) - \beta_\mathcal{I} Tr(\mathbf{B}_\mathcal{I}^T\mathbf{S}_\mathcal{I}\mathbf{B}_\mathcal{I}) \\ & + \gamma_\mathcal{I}\|\mathbf{B}_\mathcal{I}^T\mathbf{B}_\mathcal{I}\|_F^2.\end{aligned} \tag{11}$$

It is obvious that the objective function in (11) is separable. Therefore it can be solved by updating each bit while holding all the others fixed, then cycling through this process. For ease of presentation, we denote $\mathbf{z} \triangleq \mathbf{B}_\mathcal{I}^k$ to represent the k-th bit of $\mathbf{B}_\mathcal{I}$ and $\bar{\mathbf{B}}_\mathcal{I}^k$ as the concatenation of all the other fixed bits $\{\mathbf{B}_\mathcal{I}^{k'} : k' \neq k\}$. Then the problem (11) turns into minimizing the following energy

Algorithm 1 Iterative Multi-View Hashing

Input: $\mathbf{X}_* \in \mathbb{R}^{d_* \times N}$, bits number K, maximum iterations $t_{\max}$, parameters α_*, β_* and γ_*.

1: *// Initialization by Canonical Correlation Analysis as in* (1)
2: $\mathbf{W}_\mathcal{I}, \mathbf{W}_\mathcal{T} \leftarrow CCA(\mathbf{X}_\mathcal{I}, \mathbf{X}_\mathcal{T}, K)$
3: $\mathbf{B}_\mathcal{I} \leftarrow \text{sgn}(\mathbf{X}_\mathcal{I}^T \mathbf{W}_\mathcal{I})$, $\mathbf{B}_\mathcal{T} \leftarrow \text{sgn}(\mathbf{X}_\mathcal{T}^T \mathbf{W}_\mathcal{T})$
4: **repeat**
5: *// Stage 1 : Training in the feature space* $\mathcal{I}$
6: **for** $t = 1, \ldots, t_{\max}$ **do**
7: Compute $\mathbf{W}_\mathcal{I}$ by solving K independent SVM classifiers and learn rotation matrix $\mathbf{R}_\mathcal{I}$ according to (9).
8: $\mathbf{W}_\mathcal{I} \leftarrow \mathbf{W}_\mathcal{I}\mathbf{R}_\mathcal{I}$.
9: Update $\mathbf{B}_\mathcal{I}$ by solving (12) with graph cut method on each bit.
10: **end for**
11: *// Stage 2 : Training in the feature space* $\mathcal{T}$
12: Solve subproblem (6) in an similar way according to step 6-10.
13: **until** convergence or max iterations reached

Output: $\mathbf{B}_\mathcal{I} \leftarrow \text{sgn}(\mathbf{X}_\mathcal{I}^T \mathbf{W}_\mathcal{I})$, $\mathbf{B}_\mathcal{T} \leftarrow \text{sgn}(\mathbf{X}_\mathcal{T}^T \mathbf{W}_\mathcal{T})$.

function

$$\begin{aligned} &\min_z \sum_{i=1}^{N} \mathcal{E}_u(z_i) + \sum_{(i,j)} \mathcal{E}_p(z_i, z_j) \\ &s.t. \quad z_i = \{1, -1\}, i = 1, \ldots, N, \end{aligned} \tag{12}$$

where $\sum_{(i,j)}$ sums all the possible pairs of (i, j) and $i \neq j$. Obviously the problem (12) is a typical graph cut problem, which aims to find a cut to minimize the energy of a graph $\mathcal{E}$ including the energy of unary term $\mathcal{E}_u$ and the energy of pairwise term $\mathcal{E}_p$.

Specifically, from the objective function (5), it is easy to see that the unary term $\mathcal{E}_u$ involves the loss term $\mathcal{L}(\mathbf{B}_\mathcal{I}, h(\mathbf{X}_\mathcal{I}))$ and $\Phi_\mathcal{I}(\mathcal{I}, \mathcal{T})$ and can be written as $\mathcal{E}_u(z_i) = \max(0, 1 - z_i h_\mathcal{I}^k(\mathbf{X}_\mathcal{I}^i)) + \alpha_\mathcal{I} \ell_i$, where $\ell_i = \frac{\sum_{j=1}^{N}(1-s_{ij})\tau(|z_i - \mathbf{B}_\mathcal{T}^{jk}| + \sum_{t \neq k}^{K} |\mathbf{B}_\mathcal{I}^{it} - \mathbf{B}_\mathcal{T}^{jt}|)}{\sum_{j=1}^{N}(1-s_{ij})}$ according to the definition of $\Omega_\mathcal{I}(\mathcal{I}, \mathcal{T})$ in (2). From $\mathcal{O}_\mathcal{I}$, we can also see that the pairwise term $\mathcal{E}_p$ only involves the within-view similarity preservation term and the independence term. Therefore with some algebraic operations, we can see that

$$\begin{aligned} & -\beta_\mathcal{I} Tr(\mathbf{B}_\mathcal{I}^T \mathbf{S}_\mathcal{I} \mathbf{B}_\mathcal{I}) + \gamma_\mathcal{I} \|\mathbf{B}_\mathcal{I}^T \mathbf{B}_\mathcal{I}\|_F^2 \\ = \ & \mathbf{z}^T (2\gamma_\mathcal{I} \bar{\mathbf{B}}_\mathcal{I}^k (\bar{\mathbf{B}}_\mathcal{I}^k)^T - \beta_\mathcal{I} \mathbf{S}_\mathcal{I}) \mathbf{z} + const \\ = \ & \mathbf{z}^T Q_\mathcal{I} \mathbf{z} + const, \end{aligned} \tag{13}$$

where $Q_\mathcal{I} = 2\gamma_\mathcal{I} \bar{\mathbf{B}}_\mathcal{I}^k (\bar{\mathbf{B}}_\mathcal{I}^k)^T - \beta_\mathcal{I} \mathbf{S}_\mathcal{I}$. Then pairwise term can be represented as $\mathcal{E}_p(z_i, z_j) = 2Q_\mathcal{I}^{ij} z_i z_j$. To better fit $S_\mathcal{I}$ with $\bar{\mathbf{B}}_\mathcal{I}^k$, we empirically set $\gamma_\mathcal{I} = \frac{1}{2(K-1)} \beta_\mathcal{I}$ to ensure that $Q_\mathcal{I}^{ii} = 0$. Based on the above observations, the traditional graph cut algorithm [1] can be adopted to solve the problem (12) efficiently.

After we update $\mathbf{W}_\mathcal{I}$ and $\mathbf{B}_\mathcal{I}$ as a solution of (5), we take the similar way to deal with subproblem (6). By alternately solving these two subproblems, the global optimization problem (4) converges soon. Overall we summarize the main procedure in Algorithm 1, which is called *Iterative Multi-View Hashing* (IMVH). Following the same setting in [24], we also take the K eigenvectors corresponding to the largest eigenvalues by CCA as in (1) to initialize $\mathbf{W}_\mathcal{I}$ and $\mathbf{W}_\mathcal{T}$. Meanwhile, we also try random initialization for IMVH. From the comparison, we find that the different initialization is not sensitive to the final results and only affects the convergence speed.

Although Algorithm 1 consists of both inner and outer iterations, empirical experimental results show that only 3 inner iterations are enough to solve the subproblem (5) or (6) in each outer iteration, where only one cycling for the sequential bit update (12) is enough to get a satisfactory solution. What's more, usually 3 outer iterations are also sufficient to obtain hashing functions with satisfactory performance. The detailed results can be found in Figure 4 (more details will be discussed in the Section 5). Overall, IMVH successfully utilizes all the useful within-view similarity and between-view correlations across all the data modalities and unifies their respective advantages in a global frame work, which leads a superior performance compared with the existing multimodal hashing methods.

5. EXPERIMENTS

In this section, we validate our proposed method IMVH on two popular publicly available multimodal benchmarks that are fully paired and labeled including:

- **Wiki**: It contains 2,866 image-text pairs, which are selected from the Wikipedia's featured articles. Each image is represented by a 128-dim SIFT feature vector. The text article is represented by the probability distribution over 10 topics , which learned by a latent Dirichlet allocation (LDA) model. Specifically, each pair is labeled with one of 10 semantic classes. This data set is publicly available [2] and has been used in [34, 33].
- **Flickr**: It consists of 186,577 image-tag pairs, which are generated by choosing 10 largest classes from the NUS data set [3]. Each image is represented by a 500-dimensional bag-of-words feature vector based on SIFT descriptors, while each tag is represented by a 1000-dimensional feature vector. This data set is publicly available and has been used in [33].

All the data points in the above data sets are centralized in each view respectively before the subsequent process.

Since IMVH is designed as a binary encoding scheme for multi-view data, we first evaluate its effectiveness on two **Cross Modal Retrieval Tasks**: (1) use an image query in the visual modal to search the relevant texts from the text database, which can be concluded as *Image Query vs. Text Database*; (2) use a text query in the textual modal to search the relevant images from the image database, which can be concluded as *Text Query vs. Image Database*. Furthermore, it is also necessary to explore how much the incorporation of multiple information source can benefit the retrieval. Therefore, we also conduct evaluations on two **Unimodal Retrieval Tasks**: use a query to search the relevant items in the same modal, which can be concluded as (1) *Image Query vs. Image Database* and (2) *Text Query vs. Text Database*.

5.1 Experimental Setting

Following the similar setting in [33], we randomly select 20% and 1% data points as queries in Wiki and Flickr data sets respectively. The remaining is used to form the database. The retrieved points with the same semantic concept of the query are regarded as true neighbors. For the faster training procedure, we randomly select 2000 data pairs (image vs. text) in each database to construct the training set.

It is a great surprise to find that our method IMVH is not sensitive to the parameters. Therefore we unify the parameters for different

[2] http://www.svcl.ucsd.edu/projects/crossmodal/
[3] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

tasks in the different data sets. We simply set the parameters $\alpha_* = 1$, $\beta_* = 2$ and $\gamma_* = \frac{1}{2(K-1)}\beta_*$ in the objection function (4). The user-specified parameters a and b in SCISD function (3) are set to be $a = 2$ and $b = 0.2K$. Furthermore, we use the widely used LIBLINEAR package [4] to learn the SVM classifiers in the step 7 in the Algorithm 1. We adopt the GCO package [4] as the graph cut solver to solve the problem (12). For simplicity, we just run 3 outer iterations in Algorithm 1. In each outer iteration, we also only use 3 inner iterations to solve the subproblem (5) or (6). Experimental results show that the above setting works very well.

During all the evaluations, we measure the performance by using Recall curve, Precision-Recall curve (PR curve) and mean Average Precision (mAP) together. Recall curve reflects the effectiveness versus time cost, while Precision-Recall curve is a overall measure considering both precision and recall. mAP score is the average precision at the ranks where recall changes. In our experiments, each point in recall curve or precision-recall curve corresponds to a hamming radius ranging from 0 to K. All of these three evaluation metrics are widely used in the image retrieval literature [28, 9].

5.2 Comparison methods

For cross-modal retrieval tasks, we evaluate and compare the following five multimodal algorithms:

- **IMVH** algorithm, which is proposed in this paper.
- **CRH** algorithm [33], which considers the joint preservation of between-view correlations based on a co-regularized boosting framework.
- **CVH** algorithm [15], which extends traditional spectral hashing in single view to the multi-view case, aiming at preserving the between-view similarity.
- **CMSSH** algorithm [2], which learns linear hash functions based on eigendecomposition and boosting.
- **PDH** algorithm [24], which aims to find the most discriminative hashing functions by enforcing the maximum margin constraints with a block coordinate descent iterative scheme.

For unimodal retrieval tasks, we also compare IMVH with four state-of-the-art hashing methods for single-view data including

- **LSH**: Locality Sensitive Hashing [7], which generates the projection directions fundamentally based on the random projection.
- **SH**: Spectral Hashing [28], which learns the hashing functions based on spectral graph partitioning.
- **ITQ**: Iterative quantization [9], which tries to learn an optimal orthogonal rotation to minimize the quantization error of mapping this data to the vertices of binary hypercube after the PCA projection.
- **ITQ+CCA**: A variation of ITQ, which incorporates the multi-view information by using Canonical Correlation Analysis (CCA) to produce the projection matrix instead of PCA.

The implementation of all the algorithms except CVH are generously provided by the authors, where CRH is implemented using C++ and CMSSH, PDH and IMVH are implemented using MATLAB. Since the code of CVH is not publicly available, we implement CVH using MATLAB by ourselves. All the experiments are conducted on a workstation with Intel Xeon(R) E7450 CPU and 256GB memory.

[4]http://vision.csd.uwo.ca/code/

Table 1: mAP comparison results on Wiki for cross-modal retrieval tasks.

Task	Method	Code Length		
		K=24	K=48	K=64
Image Query vs. Text Database	CRH	0.2020	0.2076	0.1858
	CVH	0.1753	0.1515	0.1512
	CMSSH	0.1371	0.1655	0.1637
	PDH	0.1838	0.1581	0.1581
	IMVH	**0.2504**	**0.2715**	**0.2719**
Text Query vs. Image Database	CRH	0.1161	0.1261	0.1195
	CVH	0.1310	0.1222	0.1215
	CMSSH	0.1177	0.1195	0.1156
	PDH	0.1370	0.1287	0.1287
	IMVH	**0.1948**	**0.2084**	**0.2168**

5.3 Experimental Results on Wiki

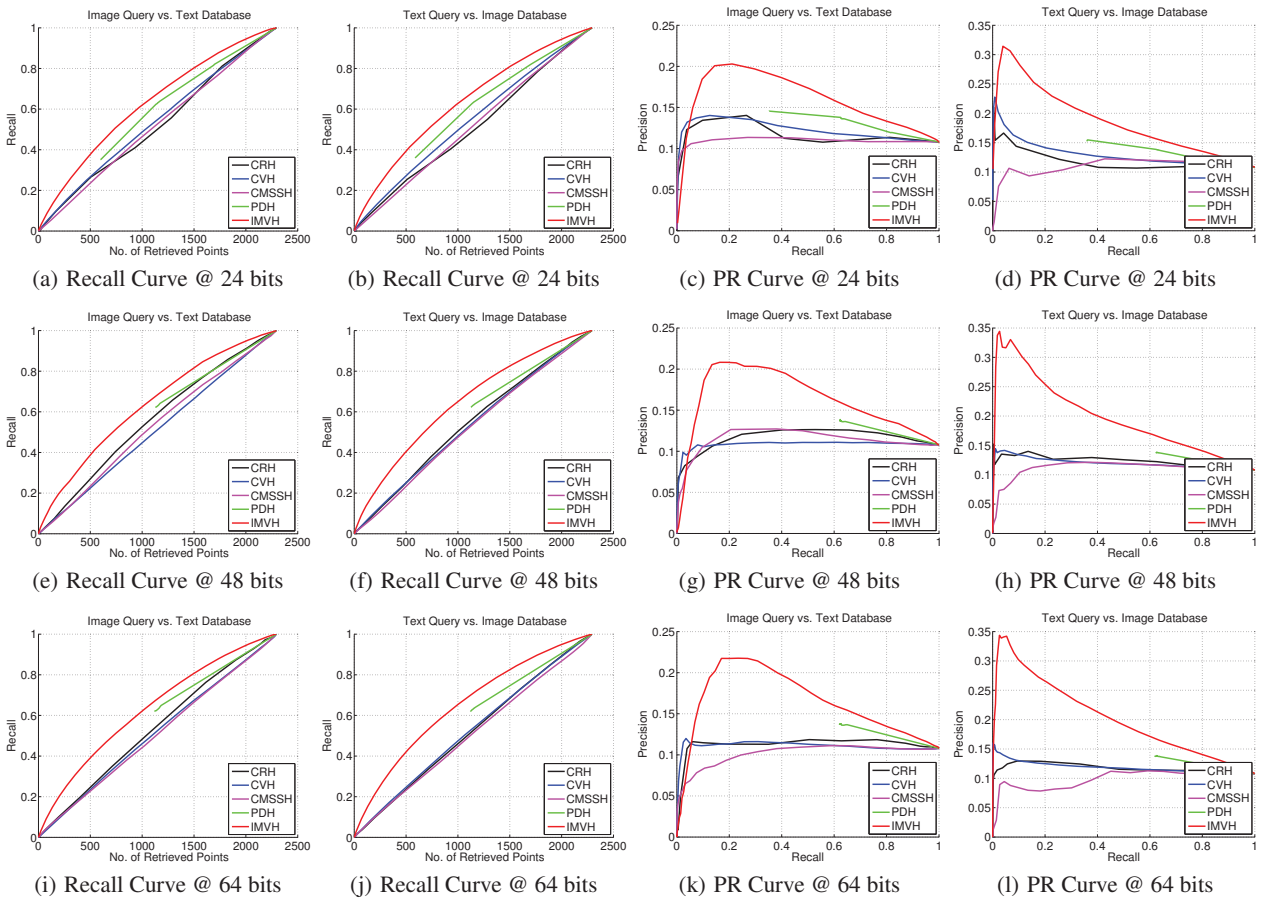
Figure 2 shows the comparison results of the cross modal retrieval tasks on the Wiki data set with different bits, which includes the recall curves and precision-recall curves for all the algorithms. The first and second columns show the performance of all the methods on recall curves for cross modal retrieval tasks *Image Query vs. Text Database* and *Text Query vs. Image Database* respectively. Given a fixed recall, the smaller of the number of the retrieved samples, the better of the algorithm, which means that the algorithm can obtain higher speed than others for the same recall. Therefore, from Figure 2, we can see that IMVH has the highest recall among all the compared methods. The third and forth columns display the comparison results on precision recall curves of all the method. Obviously, IMVH achieves significant advantage on the precision. From the above comparison results, we can see that IMVH greatly outperforms the state-of-the-art multimodal hashing methods on both cross-modal retrieval tasks. Meanwhile, we find that PDH has a second-best performance over CRH, CVH and CMSSH. However, it is important to note that the starting point of PDH's recall curve is far from zero. This is because PDH suffers from the unbalance encoding in the Wiki data set, i.e., most of the retrieved points fall into only a few hashing buckets. IMVH successfully avoids this situation by incorporating the between-view distinctiveness information, which makes a more balanced encoding scheme.

Table 1 shows the mean Average Precision (mAP) results on the Wiki data set using 24, 48, 64 bits. Obviously, the proposed method IMVH achieves higher mAP than other algorithms across all the cases. Specifically, IMVH outperforms other methods by 4.8%-8.6% in different bits for the task *Image Query vs. Text Database* and 5.7%-8.8% for the task *Text Query vs. Image Database*.

5.4 Experimental Results on Flickr

Similar to the results on the Wiki data set, IMVH also has the best performance on the Flickr data set. The recall curves and precision-recall curves are shown in the Figure 3, where IMVH still always outperforms the other multimodal methods on recall curves with a smaller gap. Similarly, IMVH consistently has the best performance of all the cases. In Table 2, we show the mAP comparison results on the Flickr where IMVH outperforms the other methods in different bits for both tasks.

To verify the claim that the incorporation of multiple information source can assist IMVH to obtain an encoding scheme with better discriminative power. We conduct a group of comparison between our method IMVH, PDH with some traditional hashing methods such as LSH, SH, ITQ and ITQ+CCA on the two uni-



(a) Recall Curve @ 24 bits (b) Recall Curve @ 24 bits (c) PR Curve @ 24 bits (d) PR Curve @ 24 bits

(e) Recall Curve @ 48 bits (f) Recall Curve @ 48 bits (g) PR Curve @ 48 bits (h) PR Curve @ 48 bits

(i) Recall Curve @ 64 bits (j) Recall Curve @ 64 bits (k) PR Curve @ 64 bits (l) PR Curve @ 64 bits

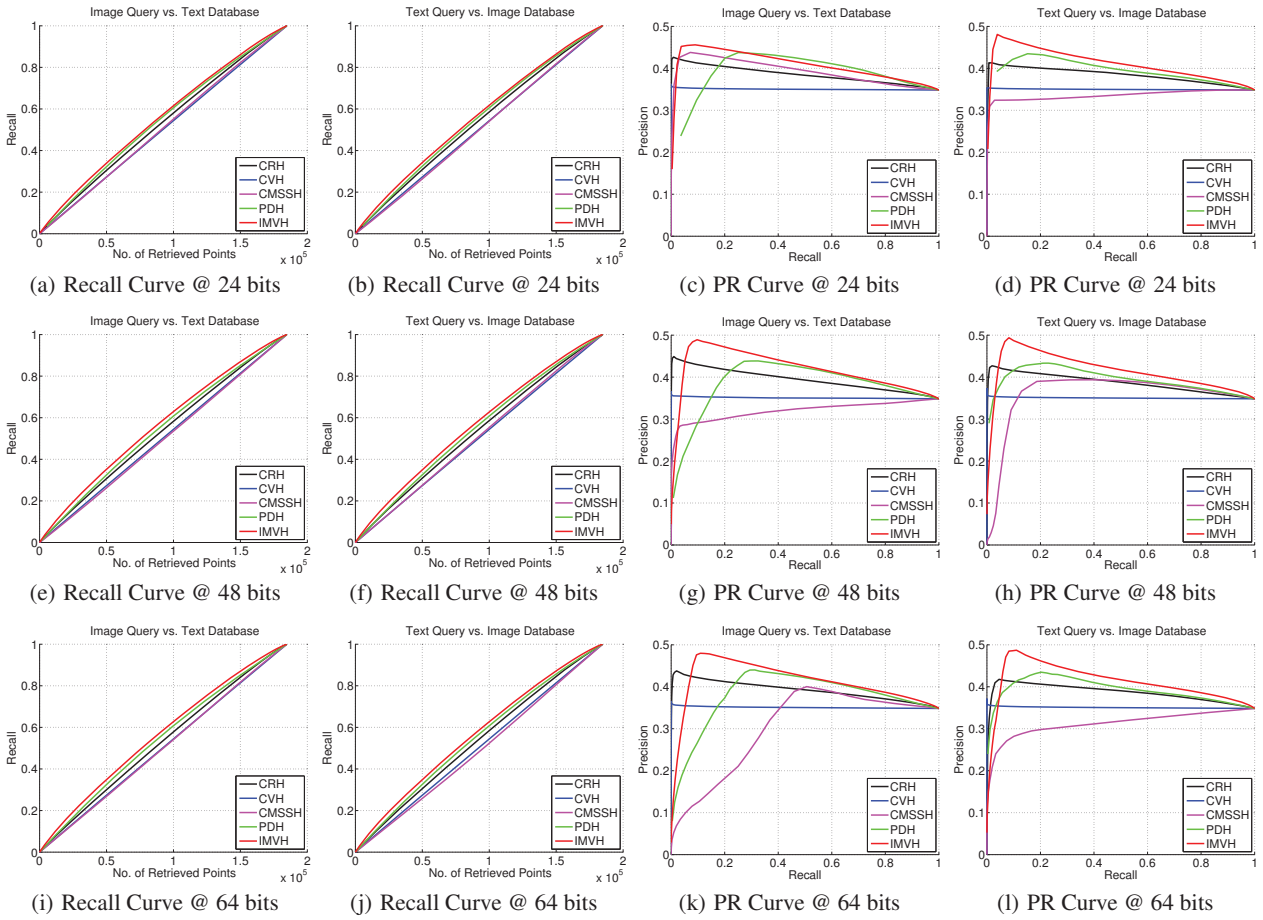
Figure 2: Cross modal retrieval results on Wiki using different bits. The first and second columns show the recall curves of all the algorithms on two cross modal retrieval tasks respectively, while the third and fourth columns show the precision-recall curves of all the algorithms on these tasks. From above results, it is obvious to see that IMVH has the best performance compared with the state-of-the-art multimodal hashing methods.

modal retrieval tasks. Note the fact that the textual feature space in Wiki data set has only 10 dimension, which is not proper for the comparison of hashing methods with more than 10 bits. Therefore we only conduct the evaluation on the Flickr data set. Among these six methods, IMVH, PDH and ITQ+CCA are trained in the multi-modal data space, while LSH, SH and ITQ are trained in the each data modal. We compare the mAP results of all the six methods in each data modality. From Figure 5, we can see that multimodal methods IMVH and PDH significantly outperform the unimodal hashing methods in both data modalities, which implies that the proper incorporation of multiple information sources would be much helpful to learn a better hashing encoding scheme. Furthermore, we can also see that IMVH always obtains a much higher mAP score than PDH with different hashing bits for the unimodal retrieval tasks in the corresponding feature spaces. Overall, from the above results, we demonstrate the effectiveness of IMVH on both unimodal and cross-modal retrieval tasks.

5.5 Convergence Rate

In this subsection, we test the convergence rate of our proposed IMVH. We conduct a series of evaluations to test the convergence rate of IMVH for both cross modal retrieval tasks on Wiki and Flickr data sets. The detailed results are shown in Figure 4. Although IMVH has both outer and inner iterations (see details in Algorithm 1), empirical experimental results show that our proposed IMVH is quite efficient. In Figure 4 (a) (b) (e) (f), we firstly fix the number of outer iterations to be 3 and evaluate the mAP score versus the number of inner iterations. We can see that only 3 inner iterations are needed to get a good enough result in a outer iteration. While in (c) (d) (g) (h), we fix the number of outer iterations to be 3 and evaluate the IMVH's performance versus the number of outer iterations, where show usually 3 outer iterations are sufficient for IMVH to obtain a satisfactory performance.

From Figure 2 and Figure 3, we can see that IMVH and PDH have the best performance during all the tasks. To be fair, we test the training time of CVH, CMSSH, PDH and IMVH since they are implemented using MATLAB while CRH is implemented using C++. From Table 3 we can see that the training time of IMVH and PDH are also far more than others. This phenomenon can be seen as as a trade off between accuracy and efficiency. For IMVH, most of the computation consumption is on solving graph cut problems (12). However, we also find that only a small training data set can guarantee the superior performance for IMVH, which can greatly reduce the computation cost caused by solving graph cut



(a) Recall Curve @ 24 bits (b) Recall Curve @ 24 bits (c) PR Curve @ 24 bits (d) PR Curve @ 24 bits

(e) Recall Curve @ 48 bits (f) Recall Curve @ 48 bits (g) PR Curve @ 48 bits (h) PR Curve @ 48 bits

(i) Recall Curve @ 64 bits (j) Recall Curve @ 64 bits (k) PR Curve @ 64 bits (l) PR Curve @ 64 bits

Figure 3: Cross modal retrieval results on Flickr using different bits. The first and second columns show the recall curves of all the algorithms on two cross modal retrieval tasks respectively, while the third and fourth columns show the precision-recall curves of all the algorithms on these tasks. From above results, we can see that IMVH consistently outperforms all the compared methods.

Table 2: mAP comparison results on Flickr for cross modal retrieval tasks.

Task	Method	Code Length		
		K=24	K=48	K=64
Image Query vs. Text Database	CRH	0.3934	0.4016	0.4015
	CVH	0.3505	0.3508	0.3510
	CMSSH	0.3838	0.3546	0.3700
	PDH	0.4250	0.4289	0.4300
	IMVH	**0.4416**	**0.4633**	**0.4658**
Text Query vs. Image Database	CRH	0.3846	0.3854	0.3859
	CVH	0.3500	0.3504	0.3506
	CMSSH	0.3432	0.3624	0.3289
	PDH	0.4063	0.4107	0.4118
	IMVH	**0.4297**	**0.4430**	**0.4444**

problem. Therefore, we can learn the discriminative hashing function by IMVH with acceptable training time even on a large data set.

5.6 Parameters' Sensitivity

Note that solving subproblem (5) or (6) involves essential parameters α_* and β_*, where α_* is used to penalize the dissimilar pairs with similar binary codes and β_* is used to control the independence of the learned hashing functions. In the previous experiments, we simply set $\alpha_* = 1$ and $\beta_* = 2$. In this subsection we conduct a series of evaluations to examine how the performance of IMVH varies with α_* and β_* separately for both cross modal retrieval tasks. The detailed results are displayed in Figure 6, where the first and second row show the respective mAP score @ 64 bits of all the algorithms on Wiki and Flickr data sets. When β_* is fixed to be 2, the impact of α_* for the final mAP score is shown in Figure 6 (a) (b) (e) (f), where we can see that IMVH can obtain consistent good performance with α_* varying from 0.2 to 20. And Figure 6 (c) (d) (g) (h) show the experimental results with different β_* when α_* is fixed to be 1, where IMVH also always has a satisfactory performance with β_* varying from 0.1 to 10. Overall, from these results it is obvious to see that our proposed IMVH is robust enough against the parameters selection.

Table 3: Training time @ 24 bits on Flickr for cross modal retrieval tasks.

Method	CVH	CMSSH	PDH	IMVH
Training time	2.5s	6.3s	297s	1917s

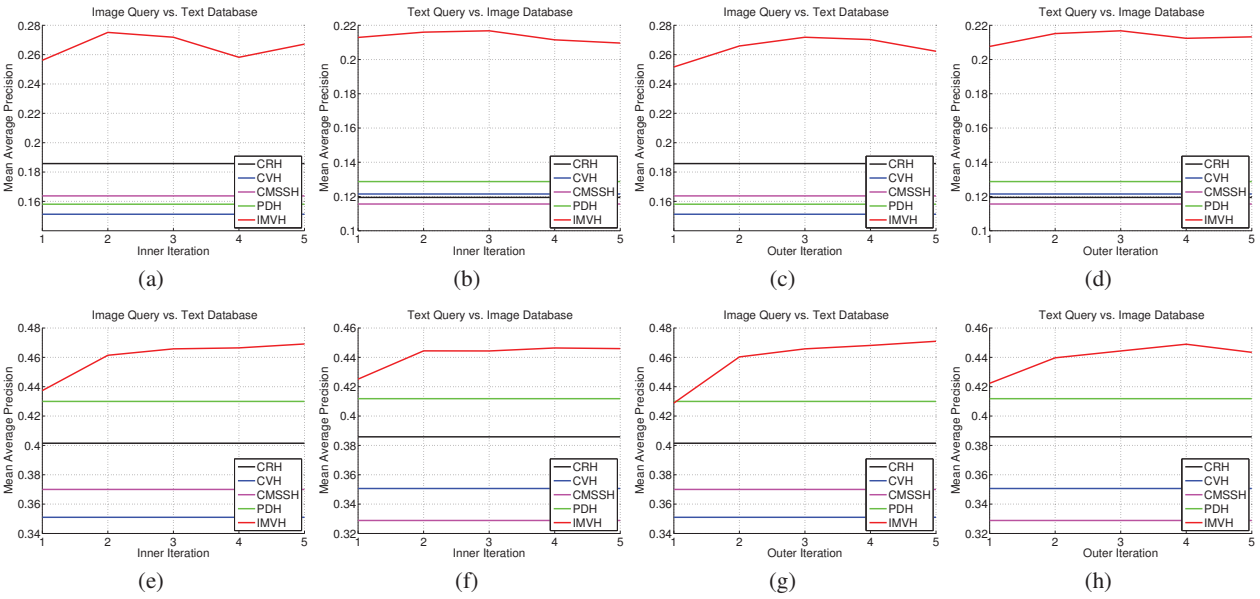


Figure 4: Iterations needed for IMVH to obtain a satisfactory performance for cross modal retrieval tasks. The first and second rows show the respective performance of IMVH on Wiki and Flickr data sets. In (a) (b) (e) (f), we fix the number of outer iterations to be 3 with varying inner iterations and in (c) (d) (g) (h) we fix the number of inner iterations to be 3 with varying outer iterations.

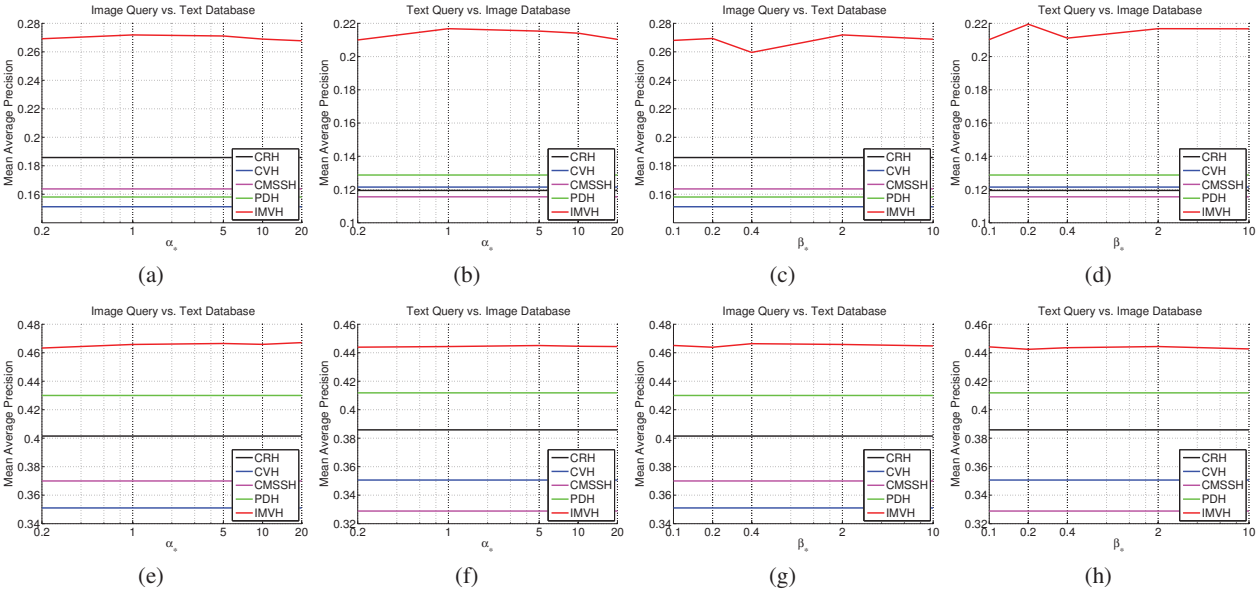


Figure 6: mAP score @ 64 bits versus the parameters β_* and α_* for both cross modal retrieval tasks. The first and second rows show the mAP score of all the algorithms on Wiki and Flickr data sets respectively. In (a) (b) (e) (f), we fix β_* to be 2 and vary α_* from 0.2 to 20. And in (c) (d) (g) (h), we fix α_* to be 1 and vary β_* from 0.1 to 10.

6. CONCLUSIONS

In this paper, we present a novel hashing algorithm for multi-view data called *Iterative Multi-View Hashing* (IMVH) for cross-modal retrieval. By incorporating the critical between-view correlations such as similarity and distinctiveness across all the views in an iterative scheme, the learned hash functions by IMVH have a greater consistent and discriminative power. To further improve the accuracy, we view the hash function learning and code generation processes separately, where the latter can be done by solving a series of binary assignment problems instead of the crucial quantization by $\text{sgn}(\cdot)$. Experimental results on two benchmarks have validated this approach, and show that this new approach outper-

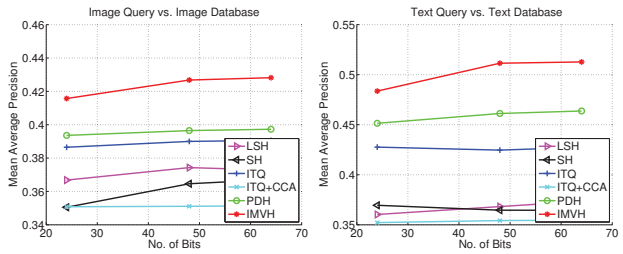


Figure 5: mAP results on Flickr for unimodal retrieval task.

forms the state-of-the-art multimodal hashing methods. In our future work, we will consider how to further improve the scalability of IMVH by exploring more efficient algorithm to solve the binary assignment problem.

7. ACKNOWLEDGEMENTS

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the National Natural Science Foundation of China under Grant 61233011, Grant 61125203, Grant 91120302, and Grant 61222207, and in part by the National Program for Special Support of Top-Notch Young Professionals. We also appreciate the constructive criticism for this work from Prof. Kaiming He in MSRA.

8. REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions Pattern Analysis Machine Intelligence*, 23(11), 2001.

[2] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Computer Vision and Pattern Recognition*, 2010.

[3] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*, 2008.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9, 2008.

[5] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Computer Vision and Pattern Recognition*, 2004.

[6] T. Ge, K. He, and J. Sun. Graph cuts for supervised hashing. In *European Conference on Computer Vision (ECCV)*, 2014.

[7] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases*, 1999.

[8] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik. Angular quantization-based binary codes for fast similarity search. In *Advances in Neural Information Processing Systems*, 2012.

[9] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions Pattern Analysis Machine Intelligence*, 35(12):2916–2929, 2012.

[10] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian. Super-bit locality-sensitive hashing. In *Advances in Neural Information Processing Systems*. 2012.

[11] Z. Jin, Y. Hu, Y. Lin, D. Zhang, S. Lin, D. Cai, and X. Li. Complementary projection hashing. In *International Conference on Computer Vision*. 2013.

[12] Z. Jin, D. Zhang, Y. Hu, S. Lin, D. Cai, and X. He. Fast and accurate hashing via iterative nearest neighbors expansion. In *IEEE Transactions on Cybernetics*. 2014.

[13] W. Kong and W.-J. Li. Isotropic hashing. In *Advances in Neural Information Processing Systems*, 2012.

[14] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, 2009.

[15] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *International Joint Conference on Artificial Intelligence*, 2011.

[16] G. Lin, C. Shen, D. Suter, and A. van den Hengel. A general two-step approach to learning-based hashing. In *International Conference on Computer Vision*, 2013.

[17] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Computer Vision and Pattern Recognition*, 2012.

[18] X. Mao, B. Lin, D. Cai, X. He, and J. Pei. Parallel field alignment for cross media retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 897–906, 2013.

[19] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions Pattern Analysis Machine Intelligence*, 29(10):1802–1817, 2007.

[20] Y. Mu, J. Shen, and S. Yan. Weakly-supervised hashing in kernel space. In *Computer Vision and Pattern Recognition*, 2010.

[21] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang. Towards cross-category knowledge propagation for learning visual concepts. In *Computer Vision and Pattern Recognition*, 2011.

[22] N. Quadrianto and C. H. Lampert. Learning multi-view neighborhood preserving projections. In *International Conference on Machine Learning*, 2011.

[23] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 251–260, 2010.

[24] M. Rastegari, J. Choi, S. Fakhraei, H. D. III, and L. S. Davis. Predictable dual-view hashing. In *International Conference on Machine Learning*, 2013.

[25] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *European Conference on Computer Vision*, 2012.

[26] P. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31, 1966.

[27] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *International Conference on Machine Learning*, 2010.

[28] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, 2005.

[29] C. Wu, J. Zhu, J. Zhang, C. Chen, and D. Cai. A convolutional treelets binary feature approach to fast keypoint recognition. In *European Conference on Computer Vision*, 2012.

[30] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang. Ranking with local regression and global alignment for cross media retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 175–184, 2009.

[31] G. Ye, D. Liu, J. Wang, and S.-F. Chang. Large-scale video hashing via structure learning. In *International Conference on Computer Vision*, 2013.

[32] D. Zhang, F. Wang, and L. Si. Composite hashing with multiple information sources. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

[33] Y. Zhen and D.-Y. Yeung. Co-regularized hashing for multimodal data. In *Advances in Neural Information Processing Systems*, 2012.

[34] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao. Linear cross-modal hashing for efficient multimedia search. In *Proceedings of the ACM International Conference on Multimedia*, pages 143–152, 2013.