Large-scale Cross-modality Search via Collective Matrix Factorization Hashing

Guiguang Ding, Yuchen Guo, Jile Zhou, and Yue Gao, *Senior Member, IEEE*

***Abstract*—By transforming data into binary representation, i.e., Hashing, we can perform high-speed search with low storage cost, and thus Hashing has collected increasing research interest in the recent years. Recently, how to generate Hashcode for multimodal data (e.g., images with textual tags, documents with photos, etc) for large-scale cross-modality search (e.g., searching semantically related images in database for a document query) is an important research issue because of the fast growth of multimodal data in the Web. To address this issue, a novel framework for multimodal Hashing is proposed, termed as Collective Matrix Factorization Hashing (CMFH). The key idea of CMFH is to learn unified Hashcodes for different modalities of one multimodal instance in the shared latent semantic space in which different modalities can be effectively connected. Therefore, accurate cross-modality search is supported. Based on the general framework, we extend it in the unsupervised scenario where it tries to preserve the Euclidean structure, and in the supervised scenario where it fully exploits the label information of data. The corresponding theoretical analysis and the optimization algorithms are given. We conducted comprehensive experiments on three benchmark datasets for cross-modality search. The experimental results demonstrate that CMFH can significantly outperform several state-of-the-art cross-modality Hashing methods, which validates the effectiveness of the proposed CMFH.**

***Index Terms*—Hashing, Multimodal Data, Collective Matrix Factorization, Cross-modality Search, Scalability, Optimization**

I. Introduction

How to index large-scale data collection for nearest neighbor (NN) retrieval is an important research topic in data engineering and database system communities. The purpose of nearest neighbor retrieval is to find the p nearest neighbors (say, 50) for a query q in a data collection with n instances based on the distance between their vectorial representations. NN retrieval plays fundamental role in several machine learning algorithms, such as kNN classifier [1], spectral clustering [2], and manifold learning [3]. Obviously, it becomes computationally costly and impractical when the retrieval is performed by computing Euclidean distance using *floating-point operation* between query and *all database instances* when n is large. Therefore, we need to design specific techniques to reduce the computational cost for NN retrieval.

One well developed paradigm is based on trees, such as kd-tree [4]. For low-dimensional data, tree can provide logarithmic complexity ($\mathcal{O}(\log n)$). However, due to the curse of dimensionality, tree will degenerate to exhaustive linear scan such that the computation is barely reduced [5]. Another celebrated paradigm is Hashing [5]. Different from trees, Hashing focuses on speeding up the operation for computing distance. Hashing transforms the data from original feature space to binary space and represents data by binary codes (i.e., Hashcodes). Given appropriate designs, the distance between original features can be well approximated by the Hamming distance (the number of different bits between binary codes). In modern CPU architecture, the Hamming distance can be computed by simple bit operations which is far faster than floating-point operations. In fact, Hashing is quite efficient for NN retrieval even though it needs to scan the database. Based on binary codes, the storage cost is significantly reduced. For example, we just need 16MB memory to store 1 million instances represented by 128-bit Hashcodes. Besides, Hashing can deal with high-dimensional data. Due to the efficiency in handling large-scale database, Hashing has drawn considerable research interest from academia and industry in recent years.

Locality Sensitive Hashing (LSH) [5] is the seminal work in Hashing. LSH has solid theoretical foundation, but it needs long Hashcodes for good performance in practice as it is *data-independent* [6]. To obtain compact Hashcodes, the information in data should be taken into consideration. Therefore, several machine learning techniques have been utilized to design *data-dependent* Hashing [7], such as Principle Components Analysis, Kmeans Clustering, Manifold Learning, Supervised Learning, Semi-supervised Learning, Deep Learning, which respectively lead to PCA Hashing [8], Kmeans Hashing [9], Spectral Hashing [10], Supervised Hashing [11], Semi-supervised Hashing [12], and Semantic Hashing [13].

Above Hashing methods focus on *unimodal* data, i.e., they can only deal with data or feature in a single type (text, image, etc). But the rapid development of Web, like news websites and social websites, has witnessed the growth of *multimodal* data, such as a news report with pictures, a photo with user-annotated tags. As mentioned in [14] and [15], *cross-modality retrieval* with multimodal data is becoming popular in recent years. The goal of cross-modality retrieval is to obtain *semantically* related data in one modality for a query in another modality. For example, with a textual query, the user hopes to get not only documents, but also some images or videos related to the query. And given an image query, it is much better if the system can return some descriptive documents related to the image than just returning some visually similar images. However, because of the heterogeneity of different feature types for different modalities (e.g., the features for text and images have different physical meaning, dimensionality, probability distribution, and etc.), it is very difficult, if not

Guiguang Ding, Yuchen Guo, Jile Zhou and Yue Gao are with the School of Software, Tsinghua University, Beijing China. Email: {dinggg, gaoyue}@tsinghua.edu.cn, {yuchen.w.guo, jile.zip}@gmail.com. Corresponding author: Guiguang Ding.

This research was supported by the National Natural Science Foundation of China (Grant No. 61571269 and 61271394).

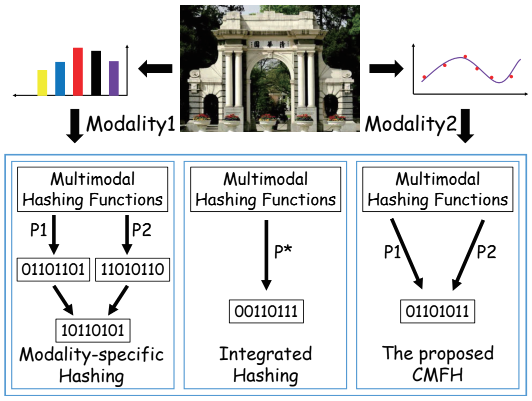

Modality1
Modality2
Multimodal Hashing Functions
P1
P2
01101101
11010110
10110101
Modality-specific Hashing
Multimodal Hashing Functions
P*
00110111
Integrated Hashing
Multimodal Hashing Functions
P1
P2
01101011
The proposed CMFH

Fig. 1: The difference among three frameworks.

impossible, to directly measure the distance (or similarity) between features. Therefore, unimodal methods mentioned above are almost infeasible for the cross-modality retrieval.

To manage the emerging multimodal data, researchers have tried some cross-media retrieval strategies, such as cross-media learning to rank [16] and mutual topic reinforce modeling [17]. Furthermore, to facilitate large-scale search, several multimodal Hashing methods have been proposed and they have achieved some promising results. Existing multimodal Hashing methods fall into the following two frameworks, *modality-specific Hashing* and *integrated Hashing*. Modality-specific Hashing (MSH) methods learn *separate* Hashcodes and the corresponding Hashing functions for different modalities of one instance. Then the Hashcodes of a multimodal instance is constructed by concatenating the Hashcodes of each modality. Besides, some specific designs are incorporated into Hashing function learning such that connection between modalities can be established for cross-modality retrieval. Because Hashing function for each modality is learned, MSH methods can construct Hashcodes for any modality. And they make efforts to connect different modalities. Hence, they can perform cross-modality retrieval. However, existing MSH methods fail to make full use of the multimodal data and only weak connection between modalities is build. On the other hand, integrated Hashing (IH) methods construct unified Hashcodes for multimodal instances and Hashing functions combining all modalities are learned. By combining more information, IH methods may achieve better retrieval performance for multimodal data. However, they require all modalities of one instance for generating Hashcodes, which is too demanding in real-world applications. In addition, they cannot generate Hashcodes for instance having partially missing modalities.

In this paper, we propose a novel framework for multimodal data, termed as **C**ollective **M**atrix **F**actorization **H**ashing (CMFH). Fig. 1 illustrates the difference between CMFH and two existing frameworks mentioned above. Specifically, CMFH assumes that all modalities of one instance should share the same Hashcodes because they have the same semantics. Hence, CMFH learns *unified* Hashcodes for different modalities of one instance in the shared latent semantic space by collective matrix factorization, which can make full use of multimodal data and effectively construct strong connection between modalities such that excellent cross-modality retrieval performance is achieved. This is intrinsically different from MSH. At the same time, the Hashing functions for each modality are also learned which can generate the final Hashcodes individually given the corresponding modality. So CMFH can handle instance with partial-missing modalities, which fixes the flaw of IH. This paper makes the following contributions

- We propose a novel framework CMFH for multimodal data which learns unified Hashcodes for different modalities of one instance in the shared latent semantic space via collective matrix factorization. CMFH can effectively connect different modalities and the learned Hashing functions for each modality can generate Hashcodes for instance with partial-missing modalities, which can result in excellent cross-modality retrieval performance. CMFH is intrinsically different from and makes significantly improvement on the existing MSH and IH frameworks.
- With the general CMFH framework, we extend the unsupervised version (UCMFH) and the supervised version (SCMFH). When labels are unavailable, UCMFH integrates a linear embedding with CMFH. Based on approximate bi-Lipschitz continuity, the learned Hashcodes can preserve the Euclidean structure in each modality. On the other hand, SCMFH incorporates some classifier-like loss into CMFH, which can fully exploit the label information.
- For both UCMFH and SCMFH, we propose effective optimization algorithms respectively. In addition, we provide corresponding theoretical analysis for them in detail.
- We carried out experiments on three benchmark datasets for cross-modality retrieval and compared CMFH with several state-of-the-art related methods. The results show the superiority of CMFH, which verifies its effectiveness.

The rest of this paper is organized as follows. We start by some preliminaries and related works in Section II. The proposed CMFH framework is introduced in Section III. In Section IV and V, we present the unsupervised and supervised versions of CMFH respectively, and the corresponding optimization algorithms and theoretical analysis are also provided in detail. Our comprehensive experiments are presented in Section VI, and at last we conclude our work in Section VII.

II. Preliminary and Related Work

A. Preliminary and Notation

A multimodal instance is denoted as $\mathbf{o}_i = \{\mathbf{x}_i^t\}$, where $\mathbf{x}_i^t \in \mathcal{X}^t$ is from the t-th modality (feature space). We generate Hashcodes for it as $\mathbf{b}_i = \{\mathbf{b}_i^t\}$, where $\mathbf{b}_i^t \in \{-1,1\}^k$ and k is the length of Hashcodes[1]. As introduced above, MSH methods obtain the multimodal Hashcodes by concatenating the Hashcode of each modality, i.e., $\mathbf{b}_i = [\mathbf{b}_i^1, ..., \mathbf{b}_i^t]$, while IH methods generate unified Hashcodes, i.e., $\mathbf{b}_i = \mathbf{b}_i^1 = \mathbf{b}_i^t$. For out-of-sample data[2], MSH methods learn Hashing function

[1]In practical implementation, we use $\{0,1\}$ as Hashcodes. In fact, these two representations are intrinsically equivalent. The transformation is achieved by setting -1 to 0. Here we use $\{-1,1\}$ for the convenience of discussion.

[2]The out-of-sample data is the one not in the training set. For generating their Hashcodes, we need to learn explicit Hashing functions when training.

for each modality individually, i.e., $\mathbf{b}_i^t = h^t(\mathbf{x}_i^t)$, while IH methods generate Hashcodes by adopting a unified Hashing function with all modalities as input, i.e., $\mathbf{b}_i = h(\mathbf{x}_i^1, ..., \mathbf{x}_i^t)$. Without loss of generality, hereafter we assume the data is zero-centered, i.e., we have $\sum_{i=1}^n \mathbf{x}_i^t = \mathbf{0}$ for any modality t.

The cross-modality retrieval can be described as follows. We have a database storing the Hashcodes of instances from one modality $\{\mathbf{b}_1^{t_1}, ..., \mathbf{b}_n^{t_1}\}$. Then given a query from another modality $\mathbf{x}_q^{t_2}$ where $t_1 \neq t_2$, we generate its Hashcode as $\mathbf{b}_q^{t_2} = h^{t_2}(\mathbf{x}_q^{t_2})$ and then compute the Hamming distance between $\mathbf{b}_q^{t_2}$ and $\mathbf{b}_i^{t_1} (i = 1, ..., n)$. Finally, the database instances are sorted by their Hamming distance to the query and the ones with smaller distance are first returned. We can observe from the procedure that we need to take the following two problems into account at training stage. First, we need to build effective connection between modalities such that the distance computing between their Hashcodes is meaningful. One important principle here is to generate similar Hashcodes for instances with similar semantics. Second, we need to learn explicit Hashing functions for each modality in order to handle instances with partial-missing modalities. IH methods do better in the first problem while MSH methods can address the second one effectively. However, MSH and IH fail to work out both simultaneously. Some frequently used notations and their corresponding descriptions are summarized in TABLE I.

B. Modality-specific Hashing

MSH methods learn separate Hashing functions for each modality. Because of the explicit Hashing functions, MSH methods can cope with out-of-sample instances with partial-missing modalities. To connect different modalities, intra-modality and inter-modality similarity are both considered.

Cross-view Hashing (CVH) [18] designs a set of Hashing functions to preserve the similarity between instances. Given a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ where S_{ij} denotes the similarity between $\mathbf{o}_i$ and $\mathbf{o}_j$ which can be defined in a supervised or an unsupervised manner. The learning objective of CVH is to minimize the *cumulative Hamming distance* for training data,

$$\begin{aligned} \min \ \mathcal{O} &= \sum_{ij} S_{ij} \sum_{t_1} \sum_{t_2 \geq t_1} \|\mathbf{b}_i^{t_1} - \mathbf{b}_j^{t_2}\|_F^2 \\ \text{s.t.} \ & \sum_i \mathbf{b}_i^t = \mathbf{0}, \sum_i (\mathbf{b}_i^t)' \mathbf{b}_i^t = n\mathbf{I}, \forall t \end{aligned} \tag{1}$$

where $\|\cdot\|_F$ represents the Frobenius norm of a vector and the orthogonality constraint on each modality is imposed to remove redundancy in the learned Hashcodes. As 1) the above function cannot generate explicit Hashing functions, and 2) the binary constraints make the problem NP-hard [10], a linear relaxation is adopted which 1) assumes a linear projection matrix $\mathbf{W}^t \in \mathbb{R}^{d^t \times k}$, and 2) removes the binary constraints:

$$\begin{aligned} \min \mathcal{O} &= \sum_{ij} S_{ij} \sum_{t_1} \sum_{t_2 \geq t_1} \|\mathbf{x}_i^{t_1} \mathbf{W}^{t_1} - \mathbf{x}_j^{t_2} \mathbf{W}^{t_2}\|_F^2 \\ \text{s.t.} \ & \sum_i \mathbf{x}_i^t \mathbf{W}^t = \mathbf{0}, \sum_i (\mathbf{x}_i^t \mathbf{W}^t)' \mathbf{x}_i^t \mathbf{W}^t = n\mathbf{I}, \forall t \end{aligned} \tag{2}$$

Above problem can be solved by generalized eigenvalue decomposition in polynomial time [19]. For the out-of-sample data $\mathbf{x}^t$, we can generate its Hashcodes by $\mathbf{b}^t = \text{sign}(\mathbf{x}^t \mathbf{W}^t)$.

TABLE I: Notations and descriptions.

Notation	Description	Notation	Description
$\mathbf{X}$	data matrix	n	#samples
$\mathbf{Y}$	label matrix	d	#features
$\mathbf{B}$	binary Hashcode	k	#Hashcode
$\mathbf{W}, \mathbf{P}$	projection matrix	c	#class
$\mathbf{U}, \mathbf{V}$	latent factors	t	#modality
$\mathbf{S}$	similarity matrix	α, μ, γ	parameters

CVH is a representative work in MSH. Several MSH methods have the similar idea to CVH, i.e., they focus on building connection between different modalities such that Hashcodes from different modalities are comparable. Cross-modality Similarity-sensitive Hashing (CMSSH) [20] aims to maximize the *correlation* between Hashcodes from different modalities. Co-regularized Hashing (CRH) [21] considers the *intra-modality loss* and inter-modality correlation simultaneously. Both methods are solved under boosting framework. Inter-media Hashing (IMH) [14] defines the intra-modality and inter-modality *consistency*. A regularized linear regression model is adopted to preserve the consistency. In Semantic Correlation Maximization Hashing (SCMH) [15], semantic label information is utilized and the learned Hashcodes are required to preserve such information. An orthogonal learning algorithm and a sequential learning algorithm are put forward.

C. Integrated Hashing

IH methods need all modalities of one instance to generate Hashcodes. One representative work is Composite Hashing with Multiple Information Sources (CHMIS) [22]. It constructs unified Hashcodes of one instance by combining the information from all sources as $\mathbf{b} = \text{sign}(\sum_t \alpha_t \mathbf{x}^t \mathbf{W}^t)$, where α_t is the nonnegative weight for the t-th modality and $\sum_t \alpha_t = 1$. After real-value relaxation, CHMIS jointly optimizes similarity preserving and consistency as follows:

$$\begin{aligned} \min \ \mathcal{O} = & C_1 \sum_{ij} \sum_t S_{ij}^t \|\mathbf{v}_i - \mathbf{v}_j\|_F^2 \\ & + C_2 \sum_i \|\mathbf{v}_i - \sum_t \alpha_t \mathbf{x}_i^t \mathbf{W}^t\|_F^2 + \sum_t \|\mathbf{W}_t\|_F^2 \\ \text{s.t.} \ & \sum_i \mathbf{v}_i = \mathbf{0}, \sum_i \mathbf{v}_i' \mathbf{v}_i = n\mathbf{I}, \alpha_t \geq 0 \end{aligned} \tag{3}$$

where C_1 and C_2 are the weight parameters to control the trate-off between different parts, S_{ij}^t is the manifold similarity [23] between $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$, and the last regularization term is to avoid overfitting [24]. This problem is solved by an iterative strategy that optimizes one variable while fixing the others.

Multi-view Spectral Hashing (MVSH) [25] is similar to CHMIS, which combines multiple features to construct similarity matrix $\mathbf{S}$. And it considers the inner product between Hashcodes as the distance measure. CHMIS and MVSH utilize multiple features to generate unified Hashcodes for one instance. Their Hashing functions require all modalities as input, which is too demanding in real world applications.

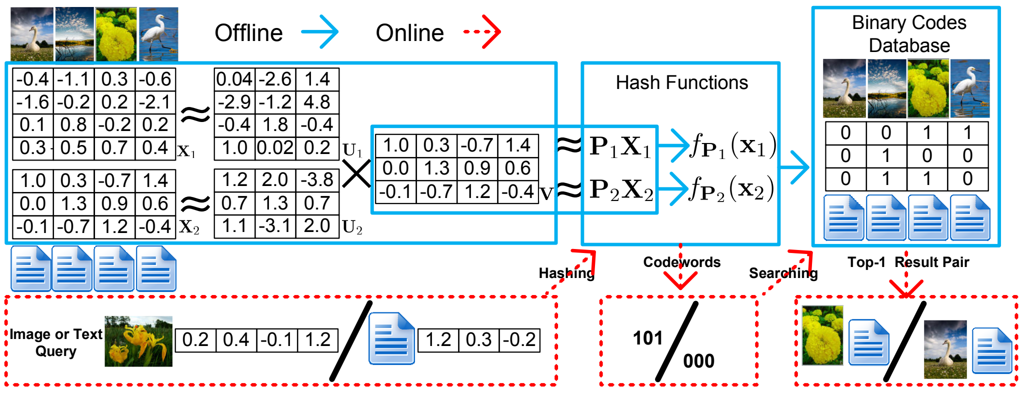

Offline
Online
Binary Codes Database
Hash Functions
$\approx \mathbf{P}_1\mathbf{X}_1 \to f_{\mathbf{P}_1}(\mathbf{x}_1)$
$\approx \mathbf{P}_2\mathbf{X}_2 \to f_{\mathbf{P}_2}(\mathbf{x}_2)$
Hashing
Codewords
Searching
Top-1 Result Pair
Image or Text Query
101
000

Fig. 2: Framework of CMFH, illustrated with two-modality toy data.

III. Collective Matrix Factorization Hashing

A. Framework Overview

The heterogeneous features from different modalities of a multimodal instance may vary a lot and they have totally different physical meaning, dimensionality, statistical properties, and etc, which makes it quite difficult to directly measure the distance between them. But fortunately, they should have the *same semantics* because they all describe the same instance. Based on this idea, we can find the shared *latent semantic space* [26] for different modalities, which is accomplished by Collective Matrix Factorization (CMF) [27] with specific regularization in this paper. Then the original heterogeneous features are projected to the shared space and finally quantified into binary codes. Since the space reflects the semantics of features and is shared by different modalities, the distance computing between modalities is feasible and meaningful. Besides, we require that the latent semantic representation (and Hashcodes) of different modalities of one instance to be *identical*, which can build strong and effective connection between modalities. And we also learn specific Hashing function for each modality such that CMFH can cope with instances with partially missing modalities, which makes it more practical.

The framework of CMFH is illustrated in Fig. 2. It consists of two parts, offline training and online coding. In offline training, we use Collective Matrix Factorization to connect different modalities and find a shared latent space for them so that the similarity can be directly measured. To generate more powerful Hashcodes, we incorporate different regularization terms under different situations. By solving the objective function, some Hashing functions are learned from training data. In this paper, we still adopt the linear function as below

$$h^t(\mathbf{x}^t) = \text{sign}(\mathbf{x}^t\mathbf{P}^t + \mathbf{a}^t) \tag{4}$$

where $\mathbf{P}^t \in \mathbb{R}^{d^t \times k}$ is a linear projection matrix and $\mathbf{a}^t$ is an offset vector to make the Hashcodes balanced. Because the input data is zero-centered, we have $\mathbf{a}^t = \mathbf{0}$. In online coding, given a query data $\mathbf{x}_q^t$ from modality t, we can generate its Hashcodes $\mathbf{b}_q^t$ by Eq. (4). Then we use $\mathbf{b}_q^t$ to retrieve data of any modalities from database based on the Hamming distance.

B. Objective Function

As we have introduced above, the key idea of the proposed CMFH is to find the shared latent semantic space and we require the Hashcodes of different modalities of one multimodal instance to be identical, i.e., we construct unified Hashcodes.

Matrix factorization (MF) techniques [26], [28] and their variants have shown effectiveness for the latent semantic analysis and achieved great success in many important works, like heperspectral unmixing [29], [30]. Specifically, given an original instance-feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, MF aims to map it to k-dimensional latent space (usually, $k < d$). Then we can obtain the latent semantic representation $\mathbf{V} \in \mathbb{R}^{n\times k}$ for instances and $\mathbf{U} \in \mathbb{R}^{k\times d}$ for features. It is shown that the similarity measure in the latent semantic space (i.e., between each row of $\mathbf{V}$) is more effective than in the original space (i.e., between each row of $\mathbf{X}$) [26], [28]. Generally, MF is achieved by minimizing the objective function shown below

$$\min_{\mathbf{U},\mathbf{V}} \ \mathcal{L}(\mathbf{X}, g(\mathbf{VU})) + \mathcal{R}(\mathbf{U},\mathbf{V}) \quad \text{s.t.} \quad \mathcal{C}(\mathbf{U},\mathbf{V}) \tag{5}$$

where $\mathcal{L}$ is a loss function, g is the prediction link, $\mathcal{R}$ is regularization on $\mathbf{U}$ and $\mathbf{V}$, $\mathcal{C}$ is the constraint on $\mathbf{U}$ and $\mathbf{V}$.

For each modality $\mathbf{X}^t$, we can factorize it to $\mathbf{U}^t$ and $\mathbf{V}^t$. However, if the factorizations are independent, we cannot connect modalities. Fortunately, it is easy to see that the "instance" is shared in each matrix $\mathbf{X}^t$ because they are different modalities of the same multimodal instances. Based on our assumption that different modalities of one multimodal instance should have the same semantics, it is reasonable to constrain that the latent representations of instances should share the latent parameters [27], i.e., $\mathbf{V}^1 = ... = \mathbf{V}^t = \mathbf{V}$, which effectively connects different modalities. With this constraint, we can jointly factorize all modalities to find the shared latent semantic space, which is formulated as follows,

$$\min_{\mathbf{U}^t,\mathbf{V}} \sum_t \alpha_t(\mathcal{L}(\mathbf{X}^t, g(\mathbf{VU}^t)) + \mathcal{R}(\mathbf{U}^t,\mathbf{V})) \text{ s.t. } \mathcal{C}(\mathbf{U}^t,\mathbf{V}) \tag{6}$$

which is the general framework of the proposed CMFH. By optimizing Eq. (6), the modality-specific Hashing functions h^t can be learned at the same time. Then we can generate Hashcodes for multimodal data for cross-modality retrieval.

Our CMFH takes advantage of the power of CMF [27] in the following aspects. 1) It turns similarity measure from original feature space to latent semantic space which better captures the *semantic* relationship between data. 2) By factorizing multimodal data jointly, a shared space is discovered, which makes distance computing between modalities feasible. To our best knowledge, we are the first the apply CMF to learn Hashing functions for cross-modality retrieval with multimodal data.

Here we formulate our CMFH to be general, in which we can choose different prediction link g, loss function $\mathcal{L}$, regularization $\mathcal{R}$ and constraint $\mathcal{C}$. Hence a user can extend it based on the specific demand. If we use identity function for g (i.e., $g(x) = x$), squared loss for $\mathcal{L}$ (i.e., $\mathcal{L}(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_F^2$), ridge regularization [31] for $\mathcal{R}$ (i.e., $\mathcal{R}(\mathbf{A}) = \|\mathbf{A}\|_F^2$) and no constraint, Eq. (6) becomes conventional CMF. However, conventional CMF cannot preserve the Euclidean structure of the features or exploit label information which are important requirements in unsupervised and supervised Hashing respectively. In the coming two sections, we will extend CMFH from the general framework in Eq. (6) to the specific formulations.

IV. Unsupervised CMFH

A. Overall Objective Function

In the unsupervised scenario, the label information is not available. So we need to preserve the Euclidean structure of original features, i.e., similar (dissimilar) instances should have similar (dissimilar) latent representations [5], which is the principle of many unsupervised Hashing methods. Formally, given two features $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ from original space, and their corresponding latent semantic representations $\mathbf{v}_i$ and $\mathbf{v}_j$, preserving the Euclidean structure can be formulated as below,

$$C_1\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F - \epsilon_1 \leq \|\mathbf{v}_i - \mathbf{v}_j\|_F \leq C_2\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F + \epsilon_2 \quad (7)$$

where C_1 and C_2 are constants. We call it approximate bi-Lipschitz continuity (ABC)[3]. The lower bound requires the latent representations of the dissimilar pair ($\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F$ is large) to be dissimilar, while the upper bound requires the ones of the similar pair ($\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F$ is small) to be similar. Now we need to incorporate proper regularization terms to Eq. (6) such that the learned representation $\mathbf{V}$ satisfies the ABC.

Graph regularization [32] is a widely used regularization for MF which can preserve the manifold structure, which has been utilized by outstanding works from several fields, like deep learning [33] and hyperspectral image destriping [34]. However, it suffers from trivial solution and scale transfer problems [35], and more importantly, it does not provide explicit function for out-of-sample data. Thus, we adopt another regularization to preserve the Euclidean structure, termed as Linear Embedding (LE), which can be formulated as below,

$$\min_{\mathbf{W}^t} \|\mathbf{V} - \mathbf{X}^t\mathbf{W}^t\|_F^2 + \mathcal{R}(\mathbf{W}^t) \quad (8)$$

Later we will prove that LE can provide the upper bound for ABC and we can easily deal with out-of-sample data with LE. Now we can incorporate LE to the general framework in Eq. (6). Specifically, we choose squared loss for $\mathcal{L}$, identity function for g, ridge regularization and no constraint, which leads to the overall objective function of UCMFH as follows

$$\min_{\mathbf{W}^t, \mathbf{U}^t, \mathbf{V}} \sum_t \alpha_t(\|\mathbf{X}^t - \mathbf{V}\mathbf{U}^t\|_F^2 + \mu\|\mathbf{V} - \mathbf{X}^t\mathbf{W}^t\|_F^2 + \gamma(\|\mathbf{U}^t\|_F^2 + \|\mathbf{W}^t\|_F^2 + \|\mathbf{V}\|_F^2)) \quad (9)$$

where α_t denotes the weight of the t-th modality, μ is the parameter to balance the trade-off between MF and LE, and γ controls the complexity of the model to avoid overfitting.

B. Optimization

Eq. (9) is non-convex with all variables $\mathbf{W}^t, \mathbf{U}^t, \mathbf{V}$ together. But fortunately, it is convex with respect to any one of them if we keep the others fixed. Therefore we can utilize an iterative algorithm for optimizing Eq. (9) until convergence as below.

Optimize $\mathbf{V}$. Denote objective function as $\mathcal{O}$, we obtain

$$\frac{\partial \mathcal{O}}{\partial \mathbf{V}} = 2\sum_t \alpha_t((\mathbf{V}\mathbf{U}^t - \mathbf{X}^t)\mathbf{U}^{t'} + \mu(\mathbf{V} - \mathbf{X}^t\mathbf{W}^t) + \gamma\mathbf{V}) \quad (10)$$

By setting $\frac{\partial \mathcal{O}}{\partial \mathbf{V}} = 0$, we obtain the updating rule for $\mathbf{V}$ below:

$$\mathbf{V} = (\sum_t \alpha_t\mathbf{X}^t(\mathbf{U}^{t'} + \mu\mathbf{W}^t))(\sum_t \alpha_t(\mathbf{U}^t\mathbf{U}^{t'} + \mu\mathbf{I} + \gamma\mathbf{I}))^{-1} \quad (11)$$

Optimize $\mathbf{U}^t$. We can observe that all modalities are decoupled when updating $\mathbf{U}^t$. Hence we can optimize $\mathbf{U}^t$ individually. The partial derivative with respective to $\mathbf{U}^t$ is

$$\frac{\partial \mathcal{O}}{\partial \mathbf{U}^t} = 2\alpha_t(\mathbf{V}'(\mathbf{V}\mathbf{U}^t - \mathbf{X}^t) + \gamma\mathbf{U}^t) \quad (12)$$

By setting $\frac{\partial \mathcal{O}}{\partial \mathbf{U}^t} = 0$, we have the updating rule for $\mathbf{U}^t$ below:

$$\mathbf{U}^t = (\mathbf{V}'\mathbf{V} + \gamma\mathbf{I})^{-1}\mathbf{V}'\mathbf{X}^t \quad (13)$$

Optimize $\mathbf{W}^t$. Analogous to $\mathbf{U}^t$, for each $\mathbf{W}^t$, we can first compute the partial derivative of $\mathcal{O}$ with respect to it as below

$$\frac{\partial \mathcal{O}}{\partial \mathbf{W}^t} = 2\alpha_t(\mu\mathbf{X}^{t'}(\mathbf{X}^t\mathbf{W}^t - \mathbf{V}) + \gamma\mathbf{W}^t) \quad (14)$$

By setting $\frac{\partial \mathcal{O}}{\partial \mathbf{W}^t} = 0$, we obtain the updating rule for $\mathbf{W}^t$ as:

$$\mathbf{W}^t = \mu(\mu\mathbf{X}^{t'}\mathbf{X}^t + \gamma\mathbf{I})^{-1}\mathbf{X}^{t'}\mathbf{V} \quad (15)$$

By iterating the steps listed above until the convergence is achieved, we can obtain the final solution. The whole training procedure of the UCMFH is summarized into Algorithm 1.

The above algorithm is for off-line training. We need to provide explicit Hashing functions for out-of-sample data, i.e., we need to learn the linear projection $\mathbf{P}^t$ for the Hashing function of each modality. Following the basic idea of CMFH, given a sample $\mathbf{x}^t$ from the t-th modality, we can obtain it latent representation $\mathbf{v}^t$ by optimizing the following problem

$$\min_{\mathbf{v}^t} \|\mathbf{x}^t - \mathbf{v}^t\mathbf{U}^t\|_F^2 + \mu\|\mathbf{v}^t - \mathbf{x}^t\mathbf{W}^t\|_F^2 + \gamma\|\mathbf{v}^t\|_F^2 \quad (16)$$

It is not difficult to derive the closed-form solution as below,

$$\mathbf{v}^t = \mathbf{x}^t(\mathbf{U}^{t'} + \mu\mathbf{W}^t)(\mathbf{U}^t\mathbf{U}^{t'} + (\mu + \gamma)\mathbf{I})^{-1} \quad (17)$$

Now we can project $\mathbf{x}^t$ to the shared space by Eq. (17). Then after a sign function, we can obtain its Hashcodes. Clearly, by comparing Eq. (17) to Eq. (4), we can obtain $\mathbf{P}^t$ as below

$$\mathbf{P}^t = (\mathbf{U}^{t'} + \mu\mathbf{W}^t)(\mathbf{U}^t\mathbf{U}^{t'} + (\mu + \gamma)\mathbf{I})^{-1} \quad (18)$$

[3]It is similar to bi-Lipschitz continuity from Mathematical Analysis, if we drop ϵ_1 and ϵ_2. Thus we call it *approximate* bi-Lipschitz continuity.

Algorithm 1 Unsupervised CMFH

```
Input:
    Training data $\mathbf{X}^t$, #Hashcode $k$, parameters $\alpha_t$, $\mu$, $\gamma$
Output:
    $\mathbf{U}^t$, $\mathbf{W}^t$, $\mathbf{V}$
1: Centralize $\mathbf{X}^t$ by the average value, $\forall t$;
2: Randomly initialize $\mathbf{U}^t$, $\mathbf{W}^t$ and $\mathbf{V}$, $\forall t$;
3: repeat
4:     Update $\mathbf{V}$ by Eq. (11);
5:     Update $\mathbf{U}^t$ by Eq. (13), $\forall t$;
6:     Update $\mathbf{W}^t$ by Eq. (15), $\forall t$;
7: until Convergence.
8: Return $\mathbf{U}^t$, $\mathbf{W}^t$, $\mathbf{V}$
```

So with Eq. (18), we have the modality-specific Hashing function h^t. Given data from any modality, we can generate Hashcodes for it by the Hashing function, and then we can perform cross-modality retrieval in a multimodal database. Besides, we can observe that $\mathbf{W}^t$ in LE can also connect $\mathbf{X}^t$ and $\mathbf{V}$, we can use $\mathbf{W}^t$ as the linear projection alternatively.

C. Analysis

The time complexity for training UCMFH is as below. The complexity for centralization (line 1) is $\mathcal{O}(ndt)$, for initialization (line 2) is $\mathcal{O}(dkt + nk)$, for updating $\mathbf{V}$ (line 4) is $\mathcal{O}(ndkt + nk + dk^2t + k^3)$, for updating $\mathbf{U}^t$ (line 5) is $\mathcal{O}((3nk^2 + k^3)t)$, for updating $\mathbf{W}^t$ (line 6) is $\mathcal{O}((2nd^2 + d^3 + ndk)t)$. Thus, the overall complexity of Algorithm 1 is $\mathcal{O}(ndt+nk+ndk+(ndkt+nk+dk^2t+k^3+(3nk^2+2nd^2+k^3+d^3+ndk)t)r)$, where r is the number of iterations to convergence. We can observe that the training time is *linear* to the size of training data n. Therefore UCMFH can easily handle large-scale training dataset. Generating Hashcodes for an out-of-sample data by Eq. (4) needs $\mathcal{O}(dk)$, which is fast.

Another important issue we need to demonstrate is that UCMFH can preserve the Euclidean structure of the original features because it satisfies ABC. Based on Eq. (9) or Eq. (16), we can see the latent representation $\mathbf{v}^t$ for data $\mathbf{x}^t$ satisfies

$$\mathbf{x}^t = \mathbf{v}^t\mathbf{U}^t + \mathbf{e}_1^t, \ \mathbf{v}^t = \mathbf{x}^t\mathbf{W}^t + \mathbf{e}_2^t \tag{19}$$

where $\mathbf{e}_1^t$ and $\mathbf{e}_2^t$ are reconstruction errors for MF and LE respectively. Based on Eq. (19), we have equations as below:

$$\begin{aligned} &\|\mathbf{x}_i^t - \mathbf{x}_j^t - (\mathbf{e}_{1i}^t - \mathbf{e}_{1j}^t)\|_F = \|(\mathbf{v}_i^t - \mathbf{v}_j^t)\mathbf{U}^t\|_F \\ &\|\mathbf{v}_i^t - \mathbf{v}_j^t\|_F = \|(\mathbf{x}_i^t - \mathbf{x}_j^t)\mathbf{W}^t + \mathbf{e}_{2i}^t - \mathbf{e}_{2j}^t\|_F \end{aligned} \tag{20}$$

By the matrix norm properties $\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F\|\mathbf{B}\|_F$ and $\|\mathbf{A}+\mathbf{B}\|_F \leq \|\mathbf{A}\|_F + \|\mathbf{B}\|_F$, we obtain inequalities as below

$$\begin{aligned} &\frac{1}{\|\mathbf{U}^t\|_F}\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F - \frac{1}{\|\mathbf{U}^t\|_F}\|\mathbf{e}_{1i}^t - \mathbf{e}_{1j}^t\|_F \\ &\leq \|\mathbf{v}_i^t - \mathbf{v}_j^t\|_F \leq \|\mathbf{W}^t\|_F\|\mathbf{x}_i^t - \mathbf{x}_j^t\|_F + \|\mathbf{e}_{2i}^t - \mathbf{e}_{2j}^t\|_F \end{aligned} \tag{21}$$

which is equivalent to Eq. (7) if we set $C_1 = 1/\|\mathbf{U}^t\|_F$, $C_2 = \|\mathbf{W}_t\|_F$, $\epsilon_1(i,j) = \|\mathbf{e}_{1i}^t - \mathbf{e}_{1j}^t\|_F/\|\mathbf{U}^t\|_F$, $\epsilon_2(i,j) = \|\mathbf{e}_{2i}^t - \mathbf{e}_{2j}^t\|_F$. Therefore UCMFH can satisfy ABC. Furthermore, we can also observe that LE provides the upper bound for ABC.

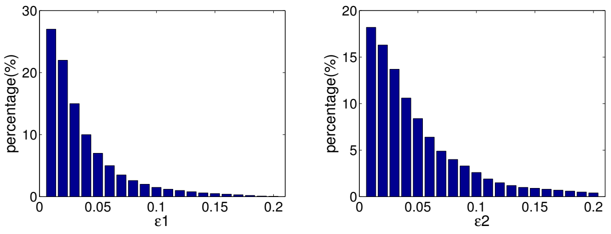


Fig. 3: The distribution of ϵ_1 and ϵ_2.

After proving UCMFH can satisfy ABC, we still need to investigate how well the approximation is, i.e., whether the lower and upper bounds are tight. If the MF and LE are perfect, i.e., $\mathbf{e}_1^t = \mathbf{e}_2^t = 0$ and $\epsilon_1 = \epsilon_2 = 0$, the bounds are tight. However, they are usually imperfect in practice, and the existence of ϵ_1 and ϵ_2 may relax the bounds, especially when they are large. For example, if ϵ_1 and ϵ_2 are infinity, the bounds will be dominated by the errors and become meaningless. Thus we wish them to be small such that tight bounds can be guaranteed. Fortunately, minimizing the objective function in Eq. (9) and Eq. (16) can reduce ϵ_1 and ϵ_2 because they aim to find $\mathbf{v}^t$ with small reconstruction error. To demonstrate this, we train 64-bit UCMFH on $5,000$ two-modality instances sampled from NUS-WIDE dataset. For each instance pair $(\mathbf{o}_i, \mathbf{o}_j)$, we compute $\epsilon_1(i,j)$ and $\epsilon_2(i,j)$ for each modality. To eliminate the influence of magnitude of input data, we normalize input data to unit Euclidean length, i.e. we consider the *relative* errors. The distribution of ϵ_1 and ϵ_2 is shown in Fig. 3. We can observe that for most pairs, the relative errors are quite small. More specifically, the relative errors of more than 90% pairs fall into $[0, 0.1]$. Such small errors indicate that the bounds in Eq. (21) are tight and UCMFH can well preserve the Euclidean structure of original features in practice. Moreover, if we keep increasing the length of the Hashcodes, the MF and LE can become more precise, which further reduces ϵ_1 and ϵ_2 and thus results in tighter bounds.

V. Supervised CMFH

A. Overall Objective Function

In supervised scenario, the semantic label information is given. Considering the goal of cross-modality retrieval is to find the *semantically* related instances, we wish that the latent representation can capture the category characteristics at the same time. To achieve this goal, one effective way is to jointly optimize the representation learning and classifier learning problems [36]. Motivated by this basic idea, we can incorporate some classifier-like loss functions with the CMFH. In general, the loss functions can be summarized as follows,

$$\min_{\mathbf{V},f} \ \sum_{j=1}^{c} \ell(\mathbf{Y}_{*j}, f_j(\mathbf{V})) + \mathcal{R}(f_j) \tag{22}$$

where f denotes a classifier, $\mathbf{Y} \in \mathbb{R}^{n\times c}$ is the label matrix where $y_{ij} = 1$ if the i-th multimodal instance belongs to the j-th category, otherwise $y_{ij} = -1$, and $\mathcal{R}(f)$ is the regularization on the classifier parameters to avoid overfitting.

Intuitively, if we use a linear classifier, minimizing above loss function will result in $\mathbf{V}$ which is separatable between categories with a hyperplane, i.e., instances from different categories will fall into different sides of a hyperplane. With this property, we can obtain similar Hashcodes for the instances from the same category but dissimilar Hashcodes for the ones belonging to different categories. Obviously, such Hashcodes can result in excellent cross-modality retrieval performance.

Now we can combine Eq. (22) with Eq. (6) for joint optimization, which leads to the objective function for SCMFH

$$\begin{aligned}\min_{\mathbf{U}^t,\mathbf{V},f} & \sum_t \alpha_t(\mathcal{L}(\mathbf{X}^t, g(\mathbf{V}\mathbf{U}^t)) + \mathcal{R}(\mathbf{U}^t,\mathbf{V})) \\ &+ \mu(\sum_{j=1}^{c} \ell(\mathbf{Y}, f_j(\mathbf{V})) + \mathcal{R}(f_j)), \text{ s.t. } \mathcal{C}(\mathbf{U}^t,\mathbf{V},f_j)\end{aligned} \tag{23}$$

Now we can choose specific settings. We use squared loss for $\mathcal{L}$, identity function for g, ridge regularization for $\mathcal{R}(\mathbf{U}^t,\mathbf{V})$. For the classifier f_j, we choose the linear form, i.e., $f_j(\mathbf{v}) = \text{sign}(\mathbf{v}\mathbf{w}_j' + b_j)$ where $\mathbf{w}_j \in \mathbb{R}^{1\times k}$ and b_j are the parameter for the classifier f_j. To make $\mathbf{V}$ more separatable, i.e., there is large margin between different categories, we utilize the SVM-like loss function [37] and constraints. We obtain the specific objective function for SCMFH as follows

$$\begin{aligned}\min_{\mathbf{U}^t,\mathbf{V},\mathbf{w}_j,b_j} & \sum_t \alpha_t(\|\mathbf{X}-\mathbf{V}\mathbf{U}^t\|_F^2 + \gamma(\|\mathbf{U}^t\|_F^2 + \|\mathbf{V}\|_F^2)) \\ &+ \mu\sum_{j=1}^{c}(\sum_i \xi_{ij} + C\|\mathbf{w}_j\|_F^2) \\ \text{s.t. } & y_{ij}(\mathbf{v}_i\mathbf{w}_j' + b_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0\end{aligned} \tag{24}$$

where α_t and μ are the weight parameters, γ controls the model complexity, and C is the trade-off parameter between maximizing margin and minimizing the classification error.

B. Optimization

Like UCMFH, the optimization problem of SCMFH can be worked out in an iterative manner until achieving convergence.

Optimize $\mathbf{w}_j$ and b_j. By fixing the others, the objective function with respect to $\mathbf{w}_j$ and b_j can be written as follows,

$$\begin{aligned}\min_{\mathbf{w}_j,b_j} \ & C\|\mathbf{w}_j\|_F^2 + \sum_i \xi_{ij} \\ \text{s.t. } & y_{ij}(\mathbf{v}\mathbf{w}_j' + b_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0\end{aligned} \tag{25}$$

This is a standard SVM training problem with $\mathbf{V}$ as input features and $\mathbf{Y}_{*j}$ as output labels. It can be effectively solved by ready-made optimization tool, such as LIBLINEAR [38].

Optimize $\mathbf{v}_i$. It is obvious that the optimization problem with respect to $\mathbf{V}$ is row-decoupled. So we can optimize each row $\mathbf{v}_i$ individually. By discarding the terms that are not relevant to $\mathbf{v}_i$, the objective function can be rewritten as

$$\begin{aligned}\min_{\mathbf{v}_i} \ \mathcal{O} = & \sum_t \alpha_t(\mathbf{v}_i\mathbf{U}^t\mathbf{U}^{t'}\mathbf{v}_i - 2\mathbf{v}_i\mathbf{U}^t\mathbf{x}_i^{t'} + \gamma\mathbf{v}_i\mathbf{v}_i') \\ &+ \mu\sum_{j=1}^{c}\xi_{ij}, \quad \text{s.t. } \ y_{ij}(\mathbf{v}\mathbf{w}_j' + b_j) \geq 1 - \xi_{ij}, \xi_{ij} \geq 0\end{aligned} \tag{26}$$

Algorithm 2 Supervised CMFH

```
Input:
    Training data $\mathbf{X}^t$, label $\mathbf{Y}$,#Hashcode $k$,
    parameters $\alpha_t$, $\mu$, $\gamma$
Output:
    $\mathbf{U}^t$, $\mathbf{V}$
1: Centralize $\mathbf{X}^t$ by the average value, $\forall t$;
2: Randomly initialize $\mathbf{U}^t$, $\mathbf{w}_j$, $b_j$ and $\mathbf{V}$, $\forall t$;
3: repeat
4:     Update $\mathbf{w}_j$ and $b_j$ by solving Eq. (25), $j = 1, ..., c$;
5:     Update $\mathbf{U}^t$ by Eq. (13), $\forall t$;
6:     Update $\mathbf{V}$ by solving Eq. (26) via Eq. (30) and (31);
7: until Convergence.
8: Return $\mathbf{U}^t$, $\mathbf{V}$
```

For convenience, we denote $\mathbf{A} = \sum_t \alpha_t(\mathbf{U}^t\mathbf{U}^{t'} + \gamma\mathbf{I})$, and $\mathbf{B} = 2\sum_t \alpha_t\mathbf{U}^t\mathbf{x}_i^{t'}$. Then denote β_j and ϕ_j as the Lagrange multipliers for the constraint. We have Lagrange $\mathcal{L}$ for $\mathcal{O}$ as

$$\begin{aligned}\mathcal{L} = & \mathbf{v}_i\mathbf{A}\mathbf{v}_i' - \mathbf{v}_i\mathbf{B} + \mu\sum_{j=1}^{c}\xi_{ij} - \sum_{j=1}^{c}\phi_j\xi_j \\ & - \sum_{j=1}^{c}\beta_j(y_{ij}(\mathbf{v}_i\mathbf{w}_j' + b_j) - 1 + \xi_{ij})\end{aligned} \tag{27}$$

Passing to the dual problem requires the two steps as below

$$\frac{\partial\mathcal{L}}{\partial\xi_j} = 0 \Rightarrow \mu - \beta_j - \phi_j = 0 \Rightarrow 0 \leq \beta_j \leq \mu \tag{28}$$

With the above identity, we formulate a reduced Lagrangian

$$\begin{aligned}\mathcal{L} &= \mathbf{v}_i\mathbf{A}\mathbf{v}_i' - \mathbf{v}_i\mathbf{B} - \sum_{j=1}^{c}\beta_j(y_{ij}(\mathbf{v}_i\mathbf{w}_j' + b_j) - 1) \\ &= \mathbf{v}_i\mathbf{A}\mathbf{v}_i' - \mathbf{v}_i\mathbf{B} - \beta\mathbf{Y}_i\mathbf{W}\mathbf{v}_i' + \beta\mathbf{g}'\end{aligned} \tag{29}$$

where $\beta = [\beta_1, ..., \beta_c]$, $\mathbf{Y}_i = \text{diag}(y_{i1}, ..., y_{ic})$, $\mathbf{W} = [\mathbf{w}_1', ..., \mathbf{w}_c']'$, and $\mathbf{g} \in \mathbb{R}^{1\times c}$ with $g_j = 1 - y_{ij}b_j$. Then taking the derivative of the reduced Lagrangian with respect to $\mathbf{v}_i$,

$$\begin{aligned}&\frac{\partial\mathcal{L}}{\partial\mathbf{v}_i} = 2\mathbf{v}_i\mathbf{A} - \mathbf{B}' - \beta\mathbf{Y}_i\mathbf{W} = 0 \\ &\Rightarrow \mathbf{v}_i = (\beta^*\mathbf{Y}_i\mathbf{W} + \mathbf{B}')\mathbf{A}^{-1}/2\end{aligned} \tag{30}$$

By substituting back in the reduced Lagrangian, we obtain

$$\beta^* = \max_{\beta} \ \beta\mathbf{H} - \beta\mathbf{M}\beta', \text{ s.t. } 0 \leq \beta_j \leq \mu \tag{31}$$

where $\mathbf{M} = \mathbf{Y}_i\mathbf{W}\mathbf{A}^{-1}\mathbf{W}'\mathbf{Y}_i'/4$ and $\mathbf{H} = -\mathbf{Y}_i\mathbf{W}\mathbf{A}^{-1}\mathbf{B}/2 + \mathbf{g}'$. Such optimization problem can be solved by quadratic programming technique. For example, we can use the Matlab function `quadprog`[4] for the optimization. The training algorithm for the SCMFH is summarized into Algorithm 2.

Optimize $\mathbf{U}^t$. When fixing the other variables, Eq. (24) is equivalent to Eq. (9). Thus we can still use Eq. (13) for $\mathbf{U}^t$.

The online coding for out-of-sample data is straightforward. Because we do not have labels for the out-of-sample data, the latent representation for $\mathbf{x}_t$ is obtained by the MF as follows

$$\min_{\mathbf{v}^t} \ \|\mathbf{x}^t - \mathbf{v}^t\mathbf{U}^t\|_F^2 + \gamma\|\mathbf{v}^t\|_F^2 \tag{32}$$

[4]http://cn.mathworks.com/help/optim/ug/quadprog.html

and the projection for modality-specific Hashing function is

$$\mathbf{P}^t = \mathbf{U}^t(\mathbf{U}^t\mathbf{U}^{t'} + \gamma\mathbf{I})^{-1} \tag{33}$$

C. Analysis

Algorithm 2 is more complicated because we adopt quadratic programming to optimize variables. Optimizing $\mathbf{w}_j$ and b_j is a standard SVM training problem, whose theoretical time complexity is $\mathcal{O}(n_{sv}^3)$ in the *worst* case, where n_{sv} is the number of support vectors which is always far smaller than n in practice. Besides, recent works, like Hash-SVM [39], have markedly improve the training efficiency and scalability of SVM. When optimizing $\mathbf{v}_i$, the most time-consuming step is the quadratic programming for solving Eq. (31). But fortunately, β has only c elements, where c is not large in real-world dataset in most cases, and the time complexity is polynomial to c. Thus, this step is also efficient. Besides, solving Eq. (31) is irrelevant to n. Therefore, the overall complexity to update $\mathbf{V}$ is linear to the number of rows, i.e., n. In summary, the time complexity of Algorithm 2 is linear to n and the number of iterations, which guarantee its scalability. In addition, the time complexity for online coding for one out-of-sample instance is $\mathcal{O}(dk)$, which is also very efficient.

VI. Experiment

A. Experiment Settings

1) Datasets: To demonstrate the efficacy of CMFH, we adopt three benchmark datasets for cross-modality retrieval.

Wiki [40]. The Wiki dataset was collected from Wikipedia webpages. It contains $2,866$ multimodal documents. Each document has 1 image and a text with at least 70 words. Each image is represented by a 128-dimensional bag-of-visual word (BoW) feature [41] based on SIFT descriptors [42], and each text is represented by a 10-dimensional topics' vector generated by the latent Dirichlet allocation (LDA) model [43]. There are 10 categories in this dataset and each multimodal document is labeled by one of them. In cross-modality retrieval, an image and a text are considered to be semantically related when they belong to the same category.

NUS-WIDE [44]. This is a large-scale dataset collected from the Web containing $269,648$ images and the corresponding tags. It contains 81 concepts but some of them are scarce. So we select the 10 most common concepts, and thus we obtain $186,577$ images. Moreover, the $1,000$ most frequent tags are selected. Each image is represented by a 500-dimensional SIFT BoW feature vector and and each text is represented by an index vector of the selected tags. Different from Wiki dataset, the image-text pair in NUS-WIDE dataset is annotated by *at least* one of the ten concepts, and pairs sharing at least one concepts are considered to be related [14].

MIRFLICKR [45]. This dataset consists of $25,000$ images annotated by some labels from 38 unique labels. Each image is described by a 100-dimensional SIFT BoW feature which mainly encodes the surface texture information, and a 144-dimensional CEDD feature [46] that focuses on color and edge directivity. SIFT and CEDD are similar in texture space but dissimilar in color space. Though this dataset contains only images, the multimodal setting can be simulated by using different visual features of images [47]. The features are considered to be related if they share at least one labels.

TABLE II: Description of benchmark datasets

Dataset	Modality	#Samples	#Feat	#Cls
Wiki	image/text	2,866	128/10	10
NUS-WIDE	image/text	186,577	500/1000	10
MIRFLICKR	SIFT/CEDD	25,000	100/144	38

To perform cross-modality retrieval, we can use one modality (like image, SIFT) as the query to retrieve data from the other modality (like text, CEDD) by the Hamming ranking. Some characteristics of the datasets are shown in TABLE II.

2) Baselines: We select several multimodal Hashing methods as baselines for comparison. Cross-view Hashing (CVH) [18], Inter-media Hashing (IMH) [14], Cross-modality Similarity Sensitive Hashing (CMSSH) [20], Quantized Correlation Hashing (QCH) [48], and Cluster-based Joint Factorization Hashing (CBH) [49]. Above five methods are unsuperivsed. Semantic Correlation Maximization Hashing [15] with sequential learning (SCMH), Kernel-based Supervised Hashing for Cross View (KSH-CV) [50], and Spectral Multimodal Hashing (SMH) [51]. Above three methods are supervised. We implement CVH and CBH by ourselves because their codes are not publicly available. For the other baseline methods, the source codes are kindly provided by their authors.

3) Metrics: Following the standard settings in literatures, like [14], [21], and etc, we adopt mean Average Precision (mAP) as the numeric evaluation metrics. mAP has shown good discriminative power and stability to evaluate the performance of cross-modality retrieval. In general, a larger mAP indicates better retrieval results that the instances which are related to the query have higher rank. Given a query and a set of R retrieved instances, the Average Precision is defined as

$$\text{AP} = \frac{1}{L}\sum\nolimits_{r=1}^{R} P(r)\delta(r) \tag{34}$$

where L is the total number of related instances in the retrieved set, $P(r)$ denotes the precision value (the ratio between the number of related instances and the number of retrieved instances) of top r retrieved instances, and $\delta(r)$ stands for an indicator function which is equal to 1 if the r-th retrieved instance is related to the query or 0 otherwise. Finally, by averaging the AP value over all queries, we obtain the mAP.

We also adopt two performance curves for evaluation, i.e., Precision-Recall (PR) curve which reflects the precision at different recall level, and precision curve which reflects the change in precision with respect to the number of retrieved instances. Both are widely adopted in Hashing evaluation [52].

4) Details: We randomly split the dataset to database and query set. Specifically, we randomly select 25%, 2% and 20% instances from Wiki, NUS-WIDE and MIRFLICKR respectively as the query set and the remained form the database. Besides, considering the large scale of NUS-WIDE and MIRFLICKR datasets, and the poor scalability of methods like CVH and IMH (their complexity is quadratic to n), we randomly select $5,000$ instances from database as the training

TABLE III: mAP Comparison on Wiki.

	Image Query v.s. Text Database				Text Query v.s. Image Database			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
IMH [14]	0.2054	0.1986	0.1997	0.2026	0.2575	0.2494	0.2367	0.2399
CVH [18]	0.2041	0.1604	0.1296	0.1308	0.2122	0.1944	0.1337	0.1125
CMSSH [20]	0.2026	0.2113	0.2011	0.2084	0.2828	0.2632	0.2653	0.2714
QCH [48]	0.2338	0.2401	0.2454	0.2456	0.3221	0.3442	0.3477	0.3495
CBH [49]	0.2199	0.2232	0.2318	0.2341	0.2974	0.3108	0.3177	0.3255
UCMFH	**0.2447**	**0.2536**	**0.2615**	**0.2652**	**0.6116**	**0.6298**	**0.6398**	**0.6477**
SCMH [15]	0.2376	0.2404	0.2472	0.2517	0.3342	0.3527	0.3604	0.3548
KSH-CV [50]	0.2241	0.2195	0.2184	0.2207	0.3075	0.3069	0.3160	0.3117
SMH [51]	0.2301	0.2375	0.2464	0.2458	0.3284	0.3592	0.3612	0.3691
SCMFH	**0.2587**	**0.2636**	**0.2801**	**0.2876**	**0.6322**	**0.6401**	**0.6574**	**0.6638**

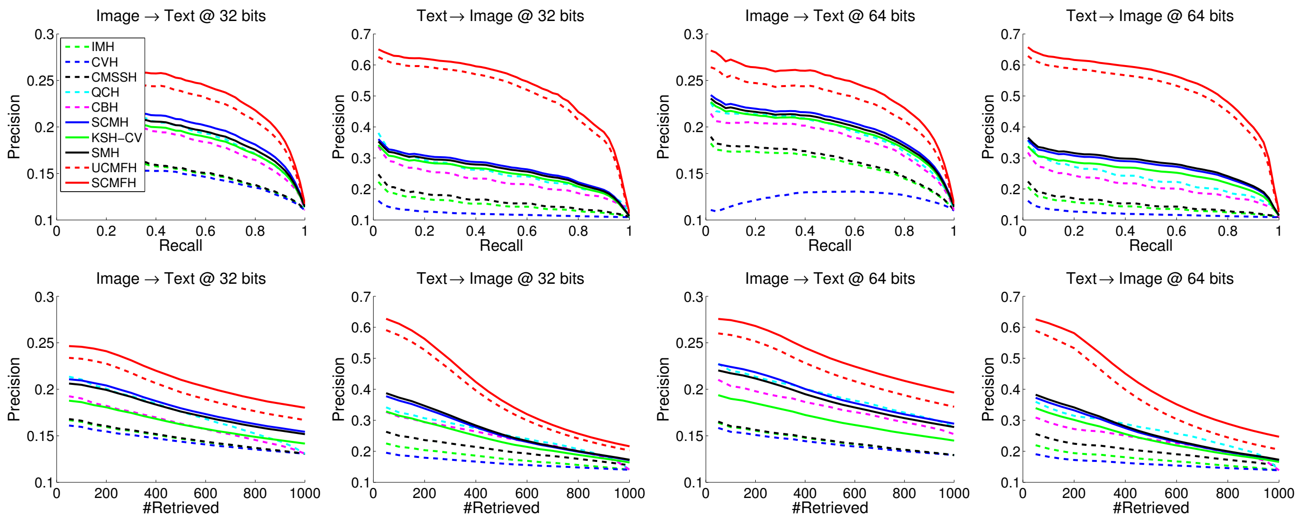


Fig. 4: Performance curves on Wiki.

set. We train Hashing functions on the training set, then generate Hashcodes for all instances in database and query set with the learned Hashcodes, and finally the retrieval is performed using the Hashcodes. This setting is also adopted in [14]. In fact, since new instances keep coming into the database all the time, the learned Hashing functions should have good ability to handle the out-of-sample data. Therefore, this setting can verify such ability of the Hashing methods.

When comparing CMFH to the baselines, we adopt the following parameter settings. For UCMFH, we set $\mu = 100$ and $\gamma = 10^{-2}$. For SCMFH, we set $\mu = 1$, $\gamma = 10^{-2}$, and $C = 1$. In the coming sections, we conduct empirical analysis on parameter sensitivity, which demonstrates that both UCMFH and SCMFH have superior and stable performance with a wide range of parameter values on all three datasets. Besides, we set $\alpha_t = 1$ for all modalities and $R = 50$ for AP.

Although the source codes for most baselines are provided, it is not fair if we just use their default parameter settings. Therefore we carefully tuned their parameters and the best results are reported. In addition, to remove the influence of any randomness caused by initialization, data split, and etc., we perform 25 repeated runs for all methods and the average results are reported. All experiments are carried out on a computer which has Intel Core i7-2600 CPU and 16GB RAM.

B. Retrieval Performance

1) Wiki: The mAP comparison on Wiki dataset with different Hashcode length is summarized in TABLE III, and the corresponding performances curves are plotted in Fig. 4. We observe that UCMFH and SCMFH markedly outperform the other baseline methods in both unsupervised and supervised scenarios. In addition, the results reveal some points below.

First, UCMFH is even superior to the supervised baselines. The experimental results indicate that connecting different modalities is a very important task in cross-modality retrieval. By learning unified Hashcodes for different modalities of one instance in the shared latent semantic space through CMFH framework, we can build strong and effective connection between modalities. On the other hand, MSH methods, such as SCMH, which can only build weak connection such that the distance computing between Hashcodes, is much less effective though supervised information is taken into consideration. This result demonstrates that the proposed *CMFH framework is superior to MSH and IH frameworks* for multimodal Hashing.

Second, both UCMFH and SCMFH have better results with longer Hashcode length. This is reasonable because longer Hashcodes can encode more information which improves the performance. Besides, with larger k, the factorization can be more precise. For UCMFH, the reconstruction errors $\mathbf{e}_1$ and $\mathbf{e}_2$ become smaller, which provides tighter lower and upper

TABLE IV: mAP Comparison on NUS-WIDE.

	Image Query v.s. Text Database				Text Query v.s. Image Database			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
IMH [14]	0.3987	0.4029	0.4015	0.3961	0.4141	0.4238	0.4256	0.4016
CVH [18]	0.4417	0.4652	0.4711	0.4649	0.4561	0.4659	0.4719	0.4821
CMSSH [20]	0.4621	0.4552	0.4597	0.4469	0.5031	0.4871	0.4853	0.4698
QCH [48]	0.5121	0.5284	0.5306	0.5381	0.5429	0.5599	0.5702	0.5721
CBH [49]	0.4802	0.4921	0.4949	0.5073	0.5033	0.5290	0.5372	0.5435
UCMFH	**0.5532**	**0.5620**	**0.5699**	**0.5813**	**0.6521**	**0.6877**	**0.7092**	**0.7177**
SCMH [15]	0.4952	0.5038	0.5212	0.5339	0.6021	0.6215	0.6304	0.6498
KSH-CV [50]	0.4687	0.4805	0.4832	0.4851	0.5118	0.5429	0.5494	0.5328
SMH [51]	0.5151	0.5277	0.5261	0.5490	0.5902	0.6133	0.6401	0.6528
SCMFH	**0.5517**	**0.5825**	**0.5974**	**0.6103**	**0.6445**	**0.6981**	**0.7268**	**0.7340**

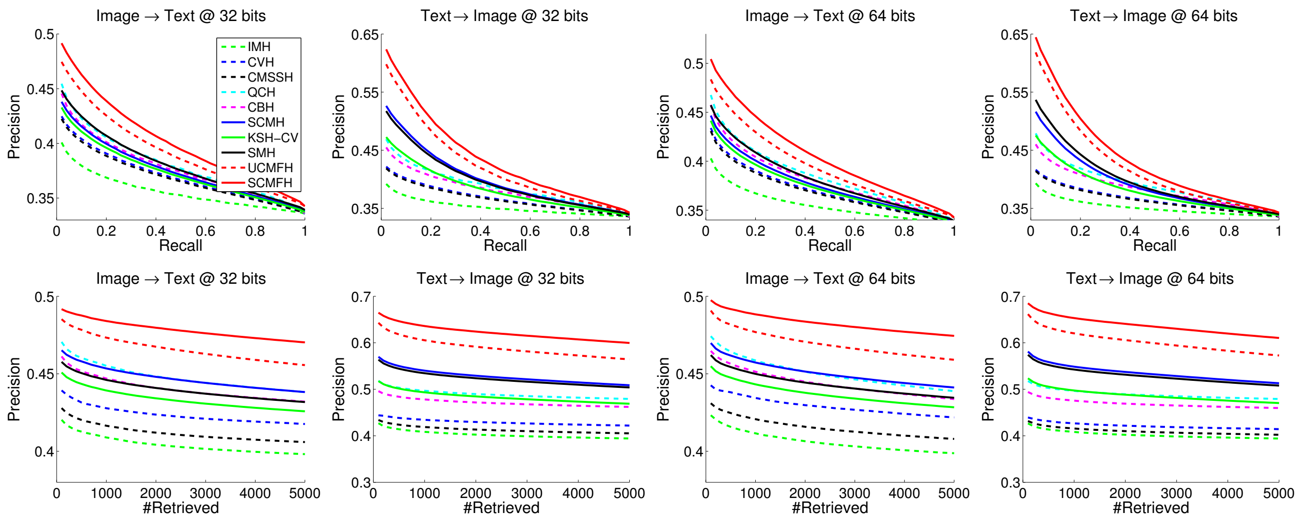


Fig. 5: Performance curves on NUS-WIDE.

bounds (smaller ϵ_1 and ϵ_2) such that the Euclidean structure can be better preserved. For SCMFH, more latent factors can provide more freedom, and thus the latent representation can better capture the category characteristics. On the other hand, we can observe the performance of some methods degrades with longer Hashcodes, like CVH. In addition, the PR curves of some methods look strange, for example, CVH for text query to image database with 64 bits performs like random guess. This phenomenon has also been observed in several literatures [21], [53], [54], [55]. In fact, these methods have orthogonality constraint on the projection directions so that each bit shows no correlation to each other. It is well known that for most real-world datasets, most of the information is contained in top few projections [55]. Thus, the first few projection directions may have higher variance and more information such that their corresponding bits are discriminative. However, with longer Hashcodes, the orthogonality constraint will force them to progressively pick the directions with little information, which leads to meaningless and ambiguous bits. Therefore, long Hashcodes are dominated by the indiscriminative bits which may markedly degrade the quality of Hashcode.

2) NUS-WIDE: The mAP of different methods is shown in TABLE IV and the corresponding performance curves are presented in Fig. 5. The results also demonstrate the superiority of UCMFH and SCMFH to the baseline methods.

In real-world database, it is always impossible to learn Hashing functions with the whole database as the training set because of the limitation of computational resource, such as memory. Besides, new data keeps coming into the database with time and it is not reasonable to produce their Hashcodes by retraining the model with all data. Moreover, in multimodal scenario, instances with partially missing modalities are very common. Therefore, a practical multimodal Hashing methods should have effective and explicit Hashing functions which can generate Hashcodes for the out-of-sample data which may have partially missing modalities. In the experiment on NUS-WIDE dataset, we select a small portion of instances from database as the training set and the Hashcodes of the whole database and the query are generated by the learned Hashing functions. This experimental setting is very similar to the real-world scenario and it can test the effectiveness of the Hashing functions. The superior performance of UCMFH and SCMFH validates that they are able to *learn effective Hashing functions for out-of-sample multimodal data*, which demonstrates their excellent ability to manage large-scale database in real world.

Another interesting observation is that the mAP improves with larger margin for text query task when we increase Hashcode length. The improvement for text query task on NUS-WIDE is 6.56% and 8.95% for UCMFH and SCMFH respectively when k increases from 16 to 128, while the

TABLE V: mAP Comparison on MIRFLICKR.

	SIFT Query v.s. CEDD Database				CEDD Query v.s. SIFT Database			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
IMH [14]	0.5874	0.5899	0.5941	0.5948	0.5704	0.5821	0.5810	0.5772
CVH [18]	0.5942	0.6017	0.6021	0.5874	0.5699	0.5748	0.5763	0.5729
CMSSH [20]	0.5856	0.5911	0.5932	0.5957	0.5602	0.5719	0.5803	0.5764
QCH [48]	0.5981	0.5988	0.6021	0.6039	0.5794	0.5802	0.5871	0.5995
CBH [49]	0.5922	0.5981	0.5926	0.5999	0.5811	0.5826	0.5849	0.5920
UCMFH	**0.6155**	**0.6221**	**0.6299**	**0.6315**	**0.6424**	**0.6563**	**0.6649**	**0.6697**
SCMH [15]	0.6142	0.6187	0.6197	0.6234	0.6361	0.6422	0.6437	0.6475
KSH-CV [50]	0.5891	0.6031	0.6040	0.5978	0.5922	0.5975	0.6020	0.6037
SMH [51]	0.6002	0.6130	0.6209	0.6251	0.6205	0.6387	0.6421	0.6476
SCMFH	**0.6221**	**0.6295**	**0.6367**	**0.6433**	**0.6516**	**0.6643**	**0.6720**	**0.6792**

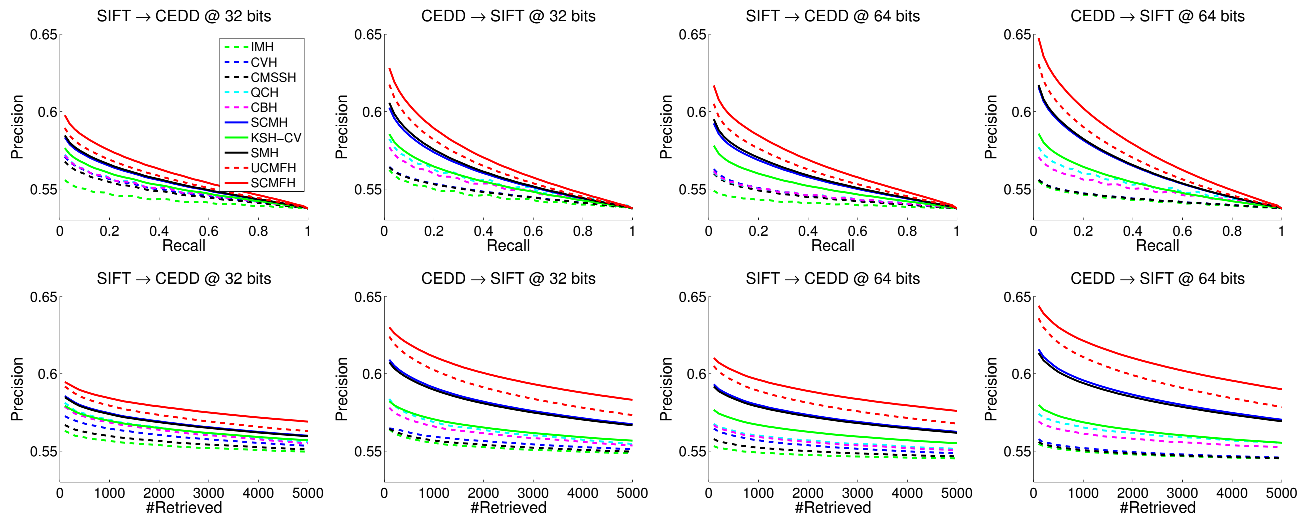


Fig. 6: Performance curves on MIRFLICKR.

improvement for the other task and datasets is much less. One possible reason is that the text feature in NUS-WIDE is high-dimensional and sparse. For such features, the matrix factorization may be imprecise with small k [56], which limits the performance of CMFH framework for short Hashcodes. Fortunately, this problem can be avoided in real-world applications. Considering the low storage cost of Hashcodes, appropriately long Hashcodes (say, more than 64 bits) are always utilized in practice. Combining our analysis for Wiki, it is obvious that the power of CMFH can be fully exploited.

3) MIRFLICKR: In TABLE V we list the mAP of different methods, and in Fig. 6 we plot the curves. Consistent with the results in the above datasets, UCMFH and SCMFH perform better than all the baseline methods with observable margin.

Combining the above experiments, we can observe that the supervised methods can achieve better performance, e.g., SCMFH v.s. UCMFH, SCMH v.s. the other baselines. In fact, our goal in cross-modality retrieve is to obtain semantically related instances. Therefore, with semantic supervision, the learned Hashing function is able to better exploit the semantic relationship between instances and help build connection between modalities, which can lead to better retrieval result.

C. Other Issues

1) Parameter Sensitivity Analysis: In both UCMFH and SCMFH, we have two important model parameters, μ as the trade-off parameter and γ to control the model complexity. We investigate how sensitive two methods are to the parameters. When analyzing one parameter, we fix the other as the value we introduced above. The mAP of two methods with respect to the different parameter values for three datasets is plotted in Fig. 7. Here we only show the results with 64-bit Hashcode. In fact, the results for the other Hashcode lengths have similar curves. We have the following observations from the results.

The weight parameter μ controls the influence of different parts on the objective function. For UCMFH, if μ is too small (say, $\mu < 0.01$), the optimization algorithm prefers to reduce the error in MF (i.e., $\mathbf{e}_1$) while it may increase the error in LE (i.e., $\mathbf{e}_2$), which will loose the upper bound in Eq. (21). If so, the Euclidean structure cannot be well preserved because similar instances may have dissimilar latent representation. On the other hand, if μ is too large (say, $\mu > 10^6$), we have tight upper bound but loose lower bound because now the algorithm focus on reducing $\mathbf{e}_2$ in LE, which makes dissimilar instances to have similar latent representation. For SCMFH, the analysis is analogous. The category characteristics cannot be well captured given a too small μ while the Hashing function will be imprecise if μ is too large. Fortunately, it is not difficult to choose proper value for μ because UCMFH performs superiorly and steadily when $\mu \in [10^{-1}, 10^3]$ and SCMFH achieves satisfactory performance during $\mu \in [10^{-1}, 10^2]$.

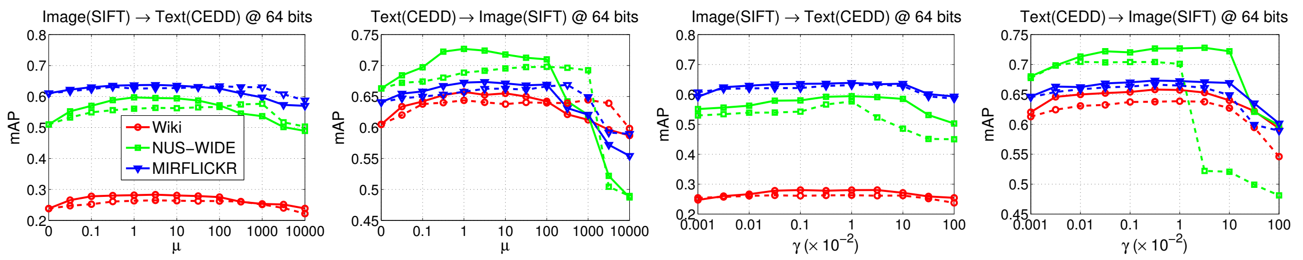


Fig. 7: Parameter Sensitivity Analysis. Dashed lines are UCMFH and solid lines are SCMFH.

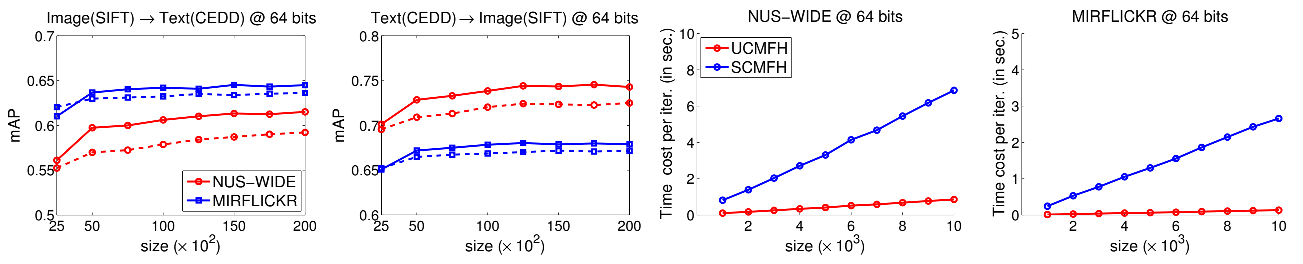
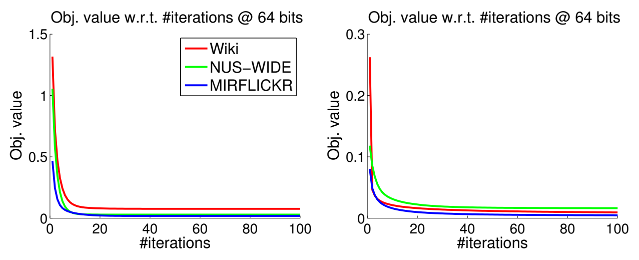


Fig. 8: Effect of training set size.



(a) UCMFH (b) SCMFH

Fig. 9: Convergence study.

The parameter γ controls the complexity of the models. If γ is too small, the model may overfit the training data and thus has large generalization error. On the other hand, if γ is too large, the model becomes under-fitted because it is required to be very simple. The results demonstrates that we can choose $\gamma \in [10^{-5}, 10^{-2}]$ for UCMFH and $\gamma \in [10^{-4}, 10^{-1}]$ for SCMFH to achieve the satisfactory and stable performance.

Moreover, we'd like to highlight the results in a special case where $\mu = 0$. In this setting, our objective function degenerates to the general CMFH. Even so, CMFH (in this situation UCMFH and SCMFH are equivalent) achieves much better results than most baseline methods, which again demonstrates the effectiveness of CMFH to build strong connection between modalities and validates the superiority of the novel CMFH framework to the MSH and IH frameworks. Besides, another important observation is that the best results of UCMFH and SCMFH are much better than the ones when $\mu = 0$. Specifically, general CMFH cannot preserve Euclidean structure in unsupervised scenario or exploit the semantic label information in supervised scenario because it only has CMF [27]. In UCMFH, we impose LE such that the Euclidean structure can be preserved. And in SCMFH, SVM-like loss function is incorporated to take the label information into consideration. The results verify that such extensions can indeed fix these flaws and make observable improvement on general CMFH.

2) *Effect of Training Set Size:* Now we investigate the effect of training set size on UCMFH and SCMFH. The first aspect is about the retrieval accuracy. On two large datasets, NUS-WIDE and MIRFLICKR, we change the size of training set and the corresponding mAP is plotted in Fig. 8. Here we can draw two conclusions. Firstly, when the training set is small (say, less than $2,500$), increasing its size can markedly improve the mAP of both UCMFH and SCMFH. This is because more training data can provide more information such that the model better captures the characteristics of the dataset which result in better performance. Secondly, keeping increasing the size after $5,000$ does not have significant influence on the result. In fact, given enough data (say, $5,000$ to $10,000$), the model is well trained and extra data is redundant. Moreover, this shows the stabilization of the Hashing function learned by the CMFH framework using reasonably a small training set.

The second is about the training time. We plot the time cost (in seconds) for each iteration of the optimization algorithms for UCMFH and SCMFH with different training set size in Fig. 8. We can observe that the cost of Algorithm 1 and 2 is *linear* to the training set size, which is consistent with our theoretically analysis. This property indicates that both algorithms are scalable and able to deal with large-scale datasets. Besides, training SCMFH is much slower than training UCMFH. This is caused by the quadratic programming to solve Eq. (31). Fortunately, we can observe the optimization is independent for each row of $\mathbf{V}$. Therefore, it is easy to use some parallel computing techniques for speeding up the training process.

3) *Convergence Study:* Since we use iterative algorithms for optimization, we empirically check the convergence property. In Fig. 9, we plot the objective function value (averaged by the number of training data) with respect to the number of

iterations. The value can decrease steadily with more iterations and can converge within 50 iterations, which validates the effectiveness of Algorithm 1 and 2. By combining the results in Fig. 8, we can see the efficiency of the Algorithm 1 and 2.

VII. Conclusion

This paper focuses on generating Hashcodes for multimodal data for cross-modality retrieval. Different from existing MSH and IH frameworks, we propose a novel CMFH framework for multimodal Hashing, whose key idea is to learn unified Hashcodes for different modalities of one instance in the shared latent semantic space. We adopt CMF to build strong connection between heterogeneous modalities. Besides, we extend CMFH framework in unsupervised scenario (UCMFH) where we integrates LE with CMFH such that the learned representation can well preserve the Euclidean structure, and in supervised scenario (SCMFH) where we impose a classifier-like loss function to exploit the semantic label information. We propose effective and efficient learning algorithms for the optimization of UCMFH and SCMFH and the theoretical analysis is also given. We conduct extensive experiment on three multimodal datasets. The cross-modality retrieval results demonstrate that both UCMFH and SCMFH can outperform the state-of-the-art Hashing methods for the multimodal data.

References

[1] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, 1992.

[2] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *JMLR*, vol. 7, pp. 2399–2434, 2006.

[4] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," in *ACM Trans. Mathematical Software*, vol. 3, no. 3, 1977, pp. 209–226.

[5] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *VLDB*, 1999, pp. 518–529.

[6] D. Zhang, J. Wang, D. Cai, and J. Lu, "Self-taught hashing for fast similarity search," in *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010, pp. 18–25.

[7] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *CoRR*, vol. abs/1606.00185, 2016.

[8] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

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

[10] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, 2008, pp. 1753–1760.

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

[12] J. Wang, O. Kumar, and S. Chang, "Semi-supervised hashing for scalable image retrieval," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3424–3431.

[13] R. Salakhutdinov and G. E. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

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

[15] D. Zhang and W. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2177–2183.

[16] F. Wu, X. Jiang, X. Li, S. Tang, W. Lu, Z. Zhang, and Y. Zhuang, "Cross-modal learning to rank via latent joint representation," *IEEE Trans. Image Processing*, vol. 24, no. 5, pp. 1497–1509, 2015.

[17] Y. Wang, F. Wu, J. Song, X. Li, and Y. Zhuang, "Multi-modal mutual topic reinforce modeling for cross-media retrieval," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 307–316.

[18] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 1360–1365.

[19] G. H. Golub and C. F. V. Loan, "Matrix computations." Johns Hopkins University Press, Baltimore, MD, 1996.

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

[21] Y. Zhen and D. Yeung, "Co-regularized hashing for multimodal data," in *26th Annual Conference on Neural Information Processing Systems 2012*, 2012, pp. 1385–1393.

[22] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011, pp. 225–234.

[23] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, 2003, pp. 153–160.

[24] V. Vapnik, "The nature of statistical learning theory." Springer Verlag, 2000.

[25] S. Kim, Y. Kang, and S. Choi, "Sequential spectral learning to hash with multiple representations," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 538–551.

[26] G. W. Furnas, S. C. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," in *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1988.

[27] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[28] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[29] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, "Manifold regularized sparse NMF for hyperspectral unmixing," *IEEE Trans. Geoscience and Remote Sensing*, vol. 51, no. 5-1, pp. 2815–2826, 2013.

[30] X. Lu, H. Wu, and Y. Yuan, "Double constrained NMF for hyperspectral unmixing," *IEEE Trans. Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2746–2758, 2014.

[31] C. M. Bishop and N. M. Nasrabadi, "*Pattern Recognition and Machine Learning*," *J. Electronic Imaging*, vol. 16, no. 4, p. 049901, 2007.

[32] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, 2011.

[33] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 10, pp. 2222–2233, 2015.

[34] X. Lu, Y. Wang, and Y. Yuan, "Graph-regularized low-rank representation for destriping of hyperspectral images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 51, no. 7-1, pp. 4009–4018, 2013.

[35] Q. Gu, C. H. Q. Ding, and J. Han, "On trivial solution and scale transfer problems in graph regularized NMF," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011.

[36] J. Feng, S. Jegelka, S. Yan, and T. Darrell, "Learning scalable discriminative dictionary with sample relatedness," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1645–1652.

[37] V. Vapnik, *Statistical learning theory*. Wiley, 1998.

[38] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[39] Y. Mu, G. Hua, W. Fan, and S. Chang, "Hash-svm: Scalable kernel machines for large-scale visual classification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 979–986.

[40] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *ACM International Conference on Multimedia*, 2010, pp. 251–260.

[41] J. Yang, Y. Jiang, A. G. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of*

the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, 2007, pp. 197–206.

[42] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[43] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[44] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “NUS-WIDE: a real-world web image database from national university of singapore,” in *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR*, 2009.

[45] M. J. Huiskes and M. S. Lew, “The mir flickr retrieval evaluation,” in *MIR ’08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*. ACM, 2008.

[46] S. A. Chatzichristofis and Y. S. Boutalis, “CEDD: color and edge directivity descriptor: A compact descriptor for image indexing and retrieval,” in *Computer Vision Systems, 6th International Conference*, 2008, pp. 312–322.

[47] N. Quadrianto and C. H. Lampert, “Learning multi-view neighborhood preserving projections,” in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 425–432.

[48] B. Wu, Q. Yang, W. Zheng, Y. Wang, and J. Wang, “Quantized correlation hashing for fast cross-modal search,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015, pp. 3946–3952.

[49] D. Rafailidis and F. Crestani, “Cluster-based joint matrix factorization hashing for cross-modal retrieval,” in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

[50] J. Zhou, G. Ding, Y. Guo, Q. Liu, and X. Dong, “Kernel-based supervised hashing for cross-view similarity search,” in *IEEE International Conference on Multimedia and Expo*, 2014, pp. 1–6.

[51] Y. Zhen, Y. Gao, D. Yeung, H. Zha, and X. Li, “Spectral multimodal hashing and its application to multimedia retrieval,” *IEEE Trans. Cybernetics*, vol. 46, no. 1, pp. 27–38, 2016.

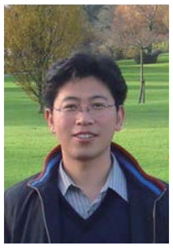
[52] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, 2013.

[53] Y. Zhen and D. Yeung, “A probabilistic model for multimodal hash function learning,” in *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 940–948.

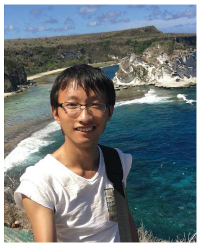
[54] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 817–824.

[55] J. Wang, S. Kumar, and S. Chang, “Semi-supervised hashing for large-scale search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, 2012.

[56] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G. Xue, Y. Yu, and Q. Yang, “Heterogeneous transfer learning for image classification,” in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2011.



Guiguang Ding received his Ph.D degree in electronic engineering from Xidian University, China. He is currently an associate professor of School of Software, Tsinghua University. Before joining school of software in 2006, he has been a postdoctoral research fellow in the Department of Automation, Tsinghua University. His current research centers on the area of multimedia information retrieval, computer vision and machine learning.



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

Jile Zhou received the B.S. degree in mathematics from Jilin University, Jilin, China, in 2011, and the M.S. degree at the School of Software, Tsinghua University, Beijing, China, in 2014. His research interests include multimedia content analysis, indexing and retrieval, and machine learning.

Yue Gao (SM’14) received his B.S. degree from Harbin Institute of Technology, Harbin, China and M.E. and Ph.D. degrees from Tsinghua University, Beijing, China, respectively.