Sparse Multi-Modal Hashing

Fei Wu, Zhou Yu, Yi Yang, Siliang Tang, Yin Zhang, and Yueting Zhuang

Abstract—Learning hash functions across heterogenous high-dimensional features is very desirable for many applications involving multi-modal data objects. In this paper, we propose an approach to obtain the sparse codesets for the data objects across different modalities via joint multi-modal dictionary learning, which we call sparse multi-modal hashing (abbreviated as SM²H). In SM²H, both intra-modality similarity and inter-modality similarity are first modeled by a hypergraph, then multi-modal dictionaries are jointly learned by Hypergraph Laplacian sparse coding. Based on the learned dictionaries, the sparse codeset of each data object is acquired and conducted for multi-modal approximate nearest neighbor retrieval using a sensitive Jaccard metric. The experimental results show that SM²H outperforms other methods in terms of mAP and Percentage on two real-world data sets.

Index Terms—Dictionary learning, multi-modal hashing, sparse coding.

I. INTRODUCTION

SIMILARITY search, a.k.a., nearest neighbor (NN) search, is a fundamental problem and has enjoyed great success in many applications of data mining, database, and information retrieval. With the explosive growth of high-dimensional data, e.g., the images and videos on the web, there is an emerging need of the NN search on high-dimensional feature space. The problem of NN search can be described as follows: given a query data object \( q \), finding the top-\( k \) nearest neighbors to the query \( q \) from a target data set.

The simplest way to solve the NN search problem is the brute-force linear search. However, this becomes prohibitively expensive when the number of retrieved target data objects are very large scale. To speed up the process of finding relevant data objects to a query, indexing techniques are necessarily conducted to organize target data objects. However, some studies pointed out that many indexing methods have an exponential dependence (in space or time or both) upon the number of dimension and even the simple brute-force linear search method may be more efficient than an index-based search in high-dimensional settings [1].

A promising way to speed up the similarity search is the hashing technique. It makes a tradeoff between accuracy and efficiency and relaxes the nearest neighbor search to approximate nearest neighbor (ANN) search. The principle of the hashing method is to map the high-dimensional data objects into compact hash codes so that similar data objects have the same or similar hash codes.

The key of the hashing-based ANN search methods is the design of the hash functions. The most well-known one is Locality Sensitive Hashing (LSH) [2], [3], which uses random projections to obtain the hash functions. However, due to the intrinsic property of random projection, to guarantee good retrieval performance, LSH usually needs a quite long hash code and hundreds of hash tables. To make the hash code more compact, several data-dependent learning based methods are proposed. Weiss et al. propose Spectral Hashing (SH) [4] which assumes the uniform distribution of data objects in training set and uses eigenfunction to obtain the hash functions. Compared with LSH, SH achieves a better performance since the learned hash functions capture the manifold of the data set. Since then, many extensions of SH have been proposed such as [5]–[8].

Nowadays, many real-world applications involve multi-modal data objects, where information inherently consists of data objects with different modalities, such as a web image with loosely related narrative text descriptions, and a historic news report with paired text and images. How to learn the latent correlation of different modalities and devise a cross-media retrieval algorithm is a very worthwhile research direction to be explored. Here, cross-media retrieval means giving a query from modality A (e.g., image modality) returning the most relevant results from modality B (e.g., textual modality). The basic idea of cross-media retrieval is to discover the correlations between paired multi-modal data and some techniques such as canonical correlation analysis [9], manifold learning [10], [11], structural learning [12] can be exploited. However, most of them do not mainly focus on the efficiency of retrieval and may be infeasible for large scale data sets. Therefore, devising one multi-modal hashing (also known as cross-media hashing) algorithm for fast cross-media retrieval is of great importance.

In the past years, only limited attempts have been made, e.g., Cross Modal Similarity Sensitive Hashing (CMSSH) [13], Cross View Hashing (CVH) [14] and Multi Latent Binary Embedding (MLBE) [15].

How to faithfully preserve both intra-modality similarity and inter-modality similarity with compact codes is fundamental for the hashing of multi-modal data objects, a natural question to ask is whether we can design an algorithm of hashing of multi-modal data objects with two following aspects: 1) the formulation of hash function may well utilize the similarities of intra-modality and inter-modality and achieve a great coding
power; and 2) the similarity of obtained hash codes should be computed efficiently.

Motivated by the fact that dictionary learning (DL) methods have the intrinsic power of dealing with the heterogeneous features by generating different dictionaries for multi-modal data objects [16], [17], this paper is dedicated to developing a hashing method for multi-modal data objects based on multi-modal dictionary learning method, i.e., simultaneous generation of the sparse coefficients for the data objects from multiple modalities (e.g., images and texts), which we call sparse multi-modal hashing (abbreviated as SM²H). SM²H is formulated by coupling the multi-modal dictionary learning (in terms of approximate reconstruction of each data object with a weighted linear combination of a small number of “basis vectors” or “dictionary atoms”) and a regularized hypergraph penalty (in terms of the modeling of multi-modal correlation).

Different from other hashing approaches, SM²H attempts to activate the most relevant component and induce a compact codeset for each data from its corresponding sparse coefficients. This characteristic enables all hashing bits to be fully utilized since each hashing bit only needs to be effective for certain data points. Although our codeset-based SM²H is different from the hamming embedding hashing methods, they actually bear some resemblance: generating a compact representation of each high-dimensional data object and performing an efficient ANN search.

To make both the similarities of intra-modality and inter-modality well preserved by the compact codesets of data objects, hypergraph is utilized to model the correlations between multi-modal data and enforced as a regularizer during multi-modal dictionary learning. As a result, the sparse coefficients of each data object can faithfully encode the intra and inter similarities of each data object with other (homogeneous or heterogeneous) data objects instead of purely reconstructive one. That is to say, similar data objects will have similar sparse coefficients. A hashing scheme is conducted to activate those informative (relevant) component indices of sparse coefficients of each data object. In this way, we can obtain a compact codeset of each data object, which provides a more compact and interpretable representation of each data object.

To the best of our knowledge, only one existing method, called Robust Sparse Hashing (RSH) [18], adopts the idea of hashing with dictionary learning. The difference between RSH and our proposed SM²H method is that RSH is limited to the uni-modal data objects while ours is performed to the multi-modal data objects. Furthermore, our method also integrates some other aforementioned appealing characteristics that make the generated sparse codesets more applicable for ANN retrieval of multi-modal data objects.

The main contributions of SM²H are two-fold:

• The intra-similarity and inter-similarity between multi-modal data objects are explicitly well leveraged when learning the multi-modal dictionaries by a hypergraph penalty which further improves the performance of ANN retrieval of multi-modal data objects.

• Since a compact codeset is learned for each data object rather than the traditional hamming binary codes in other hashing approaches, an appropriate distance metric, namely the sensitive Jaccard distance, is employed for efficient ANN search here.

The rest of the paper is organized as follows: In Section II, we review the related work of multi-modal hashing. In Section III, we give out the overview of our proposed SM²H. The optimization details are demonstrated in Section IV. Experimental results and comparisons on two real-world data sets are demonstrated in Section VI. Finally, the conclusion and future work are given in Section VII.

II. RELATED WORK

Fast finding the similar data objects to a given query from a large scale database is critical to content-based information retrieval. Given a query, the naive solution to accurately find the examples that are most similar to the query is to search over all of the data objects in database and sort them according to their similarities to the query. However, this becomes prohibitively expensive when the scale of the database is very large, therefore indexing techniques are required to accelerate the efficiency of the retrieval [19], [20].

In recent years, hashing-based methods for large-scale similarity search have sparked considerable research interests in the data mining and machine learning communities. For example, Locality Sensitive Hashing (LSH) and its variations have been proposed as indexing approaches for ANN search [21], [22]. However, LSH could be unstable and may lead to extremely bad result for a small number of hash bits and hash tables. Therefore, the number of hash bits and hash tables required may be large in some cases in order to achieve a good performance in LSH.

Unlike those approaches which randomly project the input data objects into an embedding space such as LSH, some machine learning (data-aware) approaches were recently implemented to generate more accurate hash codes, such as Semantic Hashing [23], Spectral Hashing (SH) [4], Self-taught Hashing [5], Spline Regression Hashing (SRH) [24], Random Maximum Margin Hashing [25], LDAHash [26], Bit Selection Hashing [27], Compact Hyperplane Hashing [28], etc. All of these approaches attempt to elaborate appropriate hash functions to transform original high-dimensional data objects into compact binary codes. Besides, quantization method such as Iterative quantization [29], Product quantization [30], etc., can also be adopted to conduct hashing alike efficient retrieval.

The approaches mentioned above explicitly or implicitly focus on the hashing of data objects with homogeneous features. Nevertheless, in the real world, we can extract heterogeneous features from each data object. Taken the images as examples, we can extract many of visual features from images such as global features (color, shape and texture) or local features (SIFT, Shape Context and GLOH (Gradient Location and Orientation Histogram). Therefore, we can take each kind of visual features as a view of images. Since different views (visual features) have their own specific statistical properties, different visual features may have different discriminative powers to characterize one given image. In computer vision and multimedia research, some approaches have shown that leveraging information contained in multiple views potentially has an advantage over only using a single view [31], [32]. Multiple Feature Hashing (MFH) is therefore proposed in [33], [34] to learn the binary code of each data object with heterogeneous features. MFH can preserve the local structures of each homogenous features and also globally consider the structures for all of heterogeneous features.
In this paper, we focus on performing cross-modality similarity retrieval for multi-modal data objects. In recent years, some hashing approaches for multi-modal data objects have been proposed, such as CMSSH [13], CVH [14] and MLBE [15].

The problem of multi-modal hashing has been initiated by Bronstein et al. in CMSSH [13]. Specifically, given two kinds of data objects, CMSSH learns two groups of hash functions to ensure that if two data objects (with different modalities) are relevant, their corresponding hash codes are similar and otherwise dissimilar. However, CMSSH only preserves the inter-modality similarity but ignores the intra-modality similarity.

Kumar et al. extend Spectral Hashing [4] to the multi-modal scenario and propose CVH [14]. CVH attempts to generate the hash codes by minimizing the distance of hash codes for similar data objects and maximizing the distance for dissimilar data objects. The inter-view and intra-view similarities are well conducted in CVH.

Zhen et al. propose a probabilistic latent factor model, called multi modal latent binary embedding (MLBE) in [15], to learn hash functions for multi-modal retrieval. MLBE employs a generative model to encode the intra-similarity and inter-similarity of data objects across multiple modalities. Based on maximum a posteriori estimation, the binary latent factors are efficiently obtained and then taken as hash codes in MLBE.

As stated before, this paper is interested in the sparse multi-modal hashing by multi-modal dictionary learning. Although our proposed \( \text{SM}^2 \text{H} \) bears some resemblance to robust sparse hashing (RSH) [18] that adopts the idea of hashing with dictionary learning, we extend the idea from uni-modal data objects into multi-modal data objects.

III. THE ALGORITHM OVERVIEW OF \( \text{SM}^2 \text{H} \)

In this section, we introduce the details of \( \text{SM}^2 \text{H} \). Fig. 1 illustrates the algorithmic flowchart of our proposed \( \text{SM}^2 \text{H} \). For the sake of illustrative simplicity, we assume only two kinds of data objects (i.e., images and texts) here. A hypergraph is first constructed to model the correlations between multi-modal data objects, then the multi-modal dictionaries are jointly learned to obtain one image dictionary and one text dictionary respectively. Each data object can be succinctly represented using a limited corresponding dictionary atoms and the corresponding sparse coefficients. Finally, the hashing scheme is conducted to identify those significantly informative components (i.e., the sparse codes with large coefficients). The selected component indices are used to construct a sparse codeset for each data object. We can observe the sparse codesets well preserve both intra-modality similarity and the inter-modality similarity. For examples, two “dinosaur” images have the same sparse codeset, and two “dinosaur” images have similar sparse codesets with their relevant text (dinosaur, ancient and fossil, etc.). On the contrary, two “dinosaur” images have apparently different sparse codesets with their irrelevant text (sport, football, etc.).

In Fig. 1, different kinds of data objects can be uniformly viewed as vertices in the hypergraph. The homogeneous hyperedges are utilized here to connect similar homogenous vertices, i.e., similar images or similar texts. The heterogeneous hyperedges are conducted to connect image vertices with their similar text vertices. A weight is assigned to each hyperedge according to its importance in the hypergraph. Hence, the intra-modality similarity and inter-modality similarity are well preserved in the hypergraph.

• Modeling of multi-modal correlation: In Fig. 1, different kinds of data objects can be uniformly viewed as vertices in the hypergraph. The homogeneous hyperedges are utilized here to connect similar homogenous vertices, i.e., similar images or similar texts. The heterogeneous hyperedges are conducted to connect image vertices with their similar text vertices. A weight is assigned to each hyperedge according to its importance in the hypergraph. Hence, the intra-modality similarity and inter-modality similarity are well preserved in the hypergraph.

• Multi-modal dictionary learning: The constructed hypergraph is employed as a regularizer on multi-modal dictionary learning and we can obtain one image dictionary and one text dictionary jointly. Given a data object with any modality, we can represent the data object as a weighted linear combination of a small number of corresponding “basis vectors” or “dictionary atoms”. Concretely, each data object is succinctly represented using a limited dictionary atoms and a sparse vector of weights (sparse coefficients).

• Out-of-sample extension: As we have obtained the optimal multi-modal dictionaries, we can efficiently compute the sparse coefficients for a new data object from the arbitrary modality using its corresponding dictionary.

• Hashing Scheme: Sparse coefficients of each data object is used to generate its compact representation by the hashing scheme. The hashing scheme encourages those significantly informative component indices (i.e., indices of the sparse codes with large coefficients) are selected out (activated). The selected component indices are used to construct a sparse codeset. Here, we use the sparse codeset as the hash code of each data object. Then a sensitive Jaccard distance is employed to efficiently perform ANN search.
In Fig. 1, we can observe the sparse codesets well preserve both the intra-modality similarity and the inter-modality similarity. For example, two “dinosaur” images have the same sparse codeset, and two “dinosaur” images have similar sparse codesets with their relevant text (dinosaur, ancient and fossil, etc.). On the contrary, two “dinosaur” images have apparently different sparse codesets with their irrelevant text (beef, food, lunch, etc.).

A. Notations

To simplify our presentation, we use the special case with two modalities of data objects in this paper, however, our SM2H has an inherent extension ability to more than two modalities. We name these two modalities \( X \) and \( Y \).

Assume that we have two datasets \( X \) from modality \( X \) and \( Y \) from modality \( Y \), respectively. Let \( X = [x_1, x_2, \ldots, x_{N_x}] \in \mathbb{R}^{F_x \times N_x} \) and \( Y = [y_1, y_2, \ldots, y_{N_y}] \in \mathbb{R}^{F_y \times N_y} \) be the two data matrices. \( N_x \) and \( N_y \) denote the number of data objects in \( X \) and \( Y \), respectively. \( p_x \) and \( p_y \) denote the dimensionality of two modalities (usually, \( p_x \neq p_y \)).

B. The Modeling of Multi-Modal Correlation

To well model the complex relationship (i.e., intra-similarity and inter-similarity) between \( X \) and \( Y \), we resort to the hypergraph used in [35, 36].

Let \( G(V, E, w) \) denote the weighted hypergraph where \( V \) is the set of vertices, \( E \) is the set of hyperedges, \( w \) is the weights for each hyperedge. The degree of an edge \( e \in E \) is \( \delta(e) = |e| \), that is, the cardinality of \( e \). The degree of a vertex \( d \in V \) is \( d(v) = \sum_{e \in E, v \in e} w(e) \).

Let \( D_v \in \mathbb{R}^{F \times F} \) and \( D_e \in \mathbb{R}^{|V| \times |V|} \) be two diagonal matrices consisting of the degrees of hyperedges and vertices, respectively, and \( W \in \mathbb{R}^{E \times |V|} \) be the diagonal matrix consisting of the weights of hyperedges.

The modeling of data objects with the linear combinations of a few dictionary atoms from a learned dictionary has been the focus of much recent research [38], [39]. The essential challenge to be resolved in dictionary learning is to develop an efficient approach with which each data object can be approximately reconstructed from a “optimal dictionary” with a “sparse coefficients”.

The probabilistic hypergraph encodes not only the local grouping information, but also the probability that a vertex belongs to a hyperedge. In this way, the correlation between intra-modality data objects is accurately described.

In this paper, for dataset \( X \), we first compute \( k \)-nearest neighbors of each data object in \( X \) respectively to obtain the hyperedge set \( E_x \). Then, the weight for each hyperedge \( e_i \in E_x \) is computed by \( (2) \). Finally, the incidence matrix, named as \( H_x \in \mathbb{R}^{N_x \times F_x} \), is constructed. A similar procedure is conducted to build hyperedge set \( E_y \) and incidence matrix \( H_y \in \mathbb{R}^{N_y \times F_y} \) for dataset \( Y \).

- **Homogenous hyperedges**: the intra-modality similarity across the data objects between \( X \) and \( Y \) can be encoded in hyperedge set \( E_{xy} \). Here, the elements in \( E_{xy} \) denote the inter-modality similarity between the data objects from \( X \) and \( Y \). The inter-modality similarity can be obtained according to specific applications. Since the inter-modality similarity is hard to be quantified to real values, same as [15], the binary values are used to indicate the relationships among the data objects across different modalities in this paper, i.e., if two data objects have a similar relationship, \( H_{xy} = 1 \). Otherwise, \( H_{xy} = 0 \). For examples, a web image and its corresponding loosely related narrative text can be regarded to have a similar relationship, and their similarity will be set to 1.

- **Heterogeneous hyperedges**: the inter-modality similarity across the data objects between \( X \) and \( Y \) can be encoded in hyperedge set \( E_{xy} \). Here, the elements in \( E_{xy} \) denote the inter-modality similarity between the data objects from \( X \) and \( Y \). The inter-modality similarity can be obtained according to specific applications. Since the inter-modality similarity is hard to be quantified to real values, same as [15], the binary values are used to indicate the relationships among the data objects across different modalities in this paper, i.e., if two data objects have a similar relationship, \( H_{xy} = 1 \). Otherwise, \( H_{xy} = 0 \). For examples, a web image and its corresponding loosely related narrative text can be regarded to have a similar relationship, and their similarity will be set to 1.

C. The Joint Learning of Multi-Modal Dictionaries

The modeling of data objects with the linear combinations of a few dictionary atoms from a learned dictionary has been the focus of much recent research [38], [39]. The essential challenge to be resolved in dictionary learning is to develop an efficient approach with which each data object can be approximately reconstructed from a “optimal dictionary” with “sparse coefficients”.

Suppose \( D_x = [d_1, d_2, \ldots, d_K] \in \mathbb{R}^{F_x \times K} \) is the learned dictionary from \( X \) and \( A_x = [a_1, a_2, \ldots, a_K] \in \mathbb{R}^{K \times N_x} \) is the corresponding sparse coefficients of data objects in \( X \), \( p_x \) is the dimensionality of \( X \), \( N_x \) is the number of data objects in \( X \) and \( K \) is the size of the dictionary respectively.
Similarly, \( \mathbf{D}^y = [\mathbf{d}^y_1, \mathbf{d}^y_2, \ldots, \mathbf{d}^y_K] \in \mathbb{R}^{p_y \times K} \) is the learned dictionary from \( \mathbf{Y} \) and \( \mathbf{A}^y = [a^y_1, a^y_2, \ldots, a^y_{N_y}] \in \mathbb{R}^{K \times N_y} \) indicates the corresponding sparse coefficients of data objects in \( \mathbf{Y} \), \( p_y \) is the dimensionality of \( \mathbf{Y} \), \( N_y \) is the number of data objects in \( \mathbf{Y} \) and \( K \) is the size of the dictionary respectively. Here the dictionary sizes of \( \mathbf{X} \) and \( \mathbf{Y} \) are set to a same value (i.e., \( K \)).

The objective function of learning multi-modal dictionaries can be formulated as follows:

\[
\min_{\mathbf{A}^x, \mathbf{A}^y} \| \mathbf{X} - \mathbf{D}^x \mathbf{A}^x \|_F^2 + \| \mathbf{Y} - \mathbf{D}^y \mathbf{A}^y \|_F^2 + \Omega(\mathbf{A})
\]

subject to \( \| \mathbf{d}_k^x \|_F^2 \leq 1, \| \mathbf{d}_k^y \|_F^2 \leq 1 \) for \( k = 1, 2, \ldots, K \) (3)

where \( \mathbf{A} = [\mathbf{A}^x, \mathbf{A}^y] \in \mathbb{R}^{K \times N_y} \) is the joint sparse coefficients of \( \mathbf{X} \) and \( \mathbf{Y} \) and \( N_y = N_y = N_y \).

\( \Omega(\mathbf{A}) \) shown in (3) is the imposed penalty over the sparse coefficients \( \mathbf{A} \) of all of data objects. Typically, the \( \ell_1 \)-norm \([40]\) is conducted as a penalty to explicitly encourage sparsity on each sparse coefficients \( a_i \in \mathbf{A} \) \( (i \in 1, 2, \ldots, N_y) \).

The traditional \( \ell_1 \)-norm is prone to generating the sparse coefficients of each data object independently. Therefore, the similar data objects can be encoded to have totally different sparse coefficients. Since the multi-modal correlation is encoded into the hypergraph we constructed above, it is natural to preserve the inter-modality and intra-modality similarities between sparse coefficients by imposing a hypergraph penalty in (3) \([41]\). Therefore, the \( \Omega(\mathbf{A}) \) in (3) can be defined as follows:

\[
\Omega(\mathbf{A}) = \lambda \| \mathbf{A} \|_1 + \alpha \sum_{\mathbf{r} \in \mathcal{E}} \sum_{\{i,j\} \subseteq \mathbf{r}} \frac{w(\mathbf{r})}{d(\mathbf{r})} \| a_i - a_j \|^2
\]

\[
- \lambda \| \mathbf{A} \|_1 + \alpha \text{Tr}(\mathbf{A} \mathbf{L}_h \mathbf{A}^T)
\]

where \( \text{Tr}(\cdot) \) is the trace norm of a matrix.

It can be observed that \( \Omega(\mathbf{A}) \) consists of two parts: the first part is a traditional \( \ell_1 \)-norm penalty to encourage the sparsity of sparse coefficients, and the second part ensures that for every vertex \( v_i \) and \( v_j \) in \( \mathcal{V} \) in hyperedge \( \mathbf{e} \in \mathcal{E} \), the distance of \( v_i \) and \( v_j \) is consistent with the distance of their corresponding sparse coefficients. Recall that homogenous hyperedges and heterogeneous hyperedges are both constructed, we therefore capture the intra-similarity or inter-similarity among the data objects during the multi-modal dictionary learning.

In (4), \( \mathbf{L}_h \) is called as the normalized hypergraph laplacian and defined as \([42]\):

\[
\mathbf{L}_h = \mathbf{I} - \mathbf{D}_u^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_u^{-1/2}
\]

where \( \mathbf{I} \) is the identity matrix.

According to \([41]\) and also observed in our experiments, the hypergraph laplacian regularized dictionary learning can guarantee the sparse coefficients corresponding to the data objects connected by a same hyperedge are similar to each other.

It is worth noting that the sparsity of sparse coefficients in \( \mathbf{A}^x \) and \( \mathbf{A}^y \) is likely different due to the heterogeneity in multi-modal data objects. However, in this paper, the sparsity of sparse coefficients in \( \mathbf{A}^x \) and \( \mathbf{A}^y \) is expected to be on the same level. Such similar assumption is also enforced in coupled dictionary learning (CDL) for image super-resolution in \([43]\), which suggests that one pair of image patches from different domains (low resolution v.s. high resolution) has the same dictionary atoms. It can be observed that in (4), \( \mathbf{D}_u \) and \( \mathbf{D}_e \) are set to guarantee \( \mathbf{A}^x \) and \( \mathbf{A}^y \) hold alike sparsity. In order to achieve such different level of sparsity over multi-modal data, (4) is reformulated as follows:

\[
\Omega(\mathbf{A}) = \| \mathbf{A} \|_1 + \alpha \text{Tr}(\mathbf{A} \mathbf{L}_h \mathbf{A}^T)
\]

\[
\text{subject to } \| \mathbf{d}_k^x \|_F^2 \leq 1, \| \mathbf{d}_k^y \|_F^2 \leq 1 \text{ for } k = 1, 2, \ldots, K
\]

where \( \mathbf{A} \) is a diagonal matrix defined as:

\[
\mathbf{A} = \text{diag}(\lambda_x, \ldots, \lambda_x, \lambda_y, \ldots, \lambda_y) \in \mathbb{R}^{N_y \times N_y}
\]

Since \( \mathbf{A} \) is one invariant diagonal matrix, it does not add any additional overload to the solution of (3). The detailed optimization of \( \text{SM}^2 \text{H} \) is shown in next section.

\section{D. Out-of-Sample Extension Using the Learned Dictionaries}

For each data object in the training data set, we have obtained its corresponding sparse coefficients. To compute the sparse coefficients of the \textit{out-of-sample} data objects, the learned dictionaries are exploited.

Assume that given a new data object \( x_q \) from modality \( \mathcal{X} \) (\( q \) from modality \( \mathcal{Y} \) is identical), using the dictionary \( \mathbf{D}^x \), we can obtain the sparse coefficients \( a_q \) of \( x_q \) as follows:

\[
\min_{a_q} \| x_q - \mathbf{D}^x a_q \|_2^2 + \lambda_x \| a_q \|_1
\]

An alternative strategy is to add the graph regularization term into the sparse coefficients learning like \([41]\):

\[
\min_{a_q} \| x_q - \mathbf{D}^x a_q \|_2^2 + \lambda_x \| a_q \|_1 + \alpha \sum_{i} \| a_q - u_i \|_2^2 W_i
\]

where \( N_x \) is the number of training samples from modality \( \mathcal{X} \), \( a_i \) is the sparse coefficients for \( x_i \), \( W_i \) is the similarity between \( x_q \) and \( x_i \).

Since the sparse coefficients of the \textit{out-of-sample} data objects will be online computed which is strictly restricted to the computation complexity and the time cost for (9) is about 70-100 times comparing with (8) even with a small \( K \) and \( N_x \). In our experiments, the cross-modal retrieval performance using (9) has little improvement over the one with (8), as the correlations between intra-modality and inter-modality are faithfully preserved in the learned dictionaries \( \mathbf{D}^x \) and \( \mathbf{D}^y \). Therefore, we use (8) rather than (9) to generate the sparse coefficients for the out-of-sample data objects in our experiments.

(8) is a basic \textit{lasso} problem and can be solved efficiently by LARS method \([44]\). Moreover, LARS method has a benefit that the sparse degree (number of nonzero elements) of the output \( a_q \) can be well controlled. This is helpful since we expect the sparsity of coefficients on the same level for all of multi-modal data.

\section{E. Hashing Scheme of Sparse Codesets}

Assume that the sparse coefficients for both data objects in training sets and the \textit{out-of-sample} data objects are obtained, we tend to devise a hashing scheme to generate the \textit{sparse codeset} of each data in order to perform efficient ANN search on a large-scale data set.

Kong et al. have pointed out that the quantization method of simply thresholding the linear projected data objects into binary hamming codes may lead to serious information loss \([45]\). As
a result, different from traditional hashing approaches to generate binary hamming codes, the hashing scheme we utilize in this paper is to identify those informative component indices in sparse coefficients of each data object. We propose two hashing schemes to generate a sparse codeset for each data object as follows:

Hashing Scheme I: Given a data object $x$, if $a^x = (a_1^x, a_2^x, \ldots, a_K^x) \in \mathbb{R}^K$ is its sparse coefficients learned from corresponding dictionary $D^x$ with $K$ dictionary atoms, we activate the “meaningful” components from $a^x$ to generate its sparse codeset as follows:

$$\text{SC}(a^x) = \{i \mid \forall i \in 1, 2, \ldots, K, \text{ if } |a_i^x| > \sigma \}$$

where $\sigma > 0$ is a relatively small-value threshold.

According to (10), each data object $x$ can be hashed into $\text{SC}(a^x)$, we call $\text{SC}(a^x)$ is the sparse codeset of $x$. In fact, $\text{SC}(a^x)$ is a set that consists of those indices of informative component indices in the sparse coefficients of $x$. Therefore, the traditional Jaccard similarity can be conducted to measure the similarity of two sparse coefficients $a^x$ and $a^y$.

$$\text{Sim}(a^x, a^y) = \frac{|\text{SC}(a^x) \cap \text{SC}(a^y)|}{|\text{SC}(a^x) \cup \text{SC}(a^y)|}$$

where $\cdot |$ stands for the cardinality of the set.

It is apparent that the hashing scheme in (10) has the power to select out those significant component indices.

Hashing Scheme II: Notice that the elements in the sparse coefficients $a^x$ of data object $x$ can be positive, zero, or negative. The meaning of positive elements and negative elements are somehow different. We use $\text{SC}_+(a^x)$ and $\text{SC}_-(a^x)$ to respectively denote those meaningful positive bits and negative bits in $a^x$ as follows:

$$\text{SC}_+(a^x) = \{i \mid \forall i \in 1, 2, \ldots, K, \text{ if } a_i^x > \sigma \}$$

$$\text{SC}_-(a^x) = \{i \mid \forall i \in 1, 2, \ldots, K, \text{ if } a_i^x < -\sigma \}$$

(12)

Now, each data object $x$ can be represented as two sparse codesets: $\text{SC}_+(\cdot)$ and $\text{SC}_-(\cdot)$. The sign-sensitive similarity of two sparse coefficients $a^x$ and $a^y$ can be defined as follows:

$$\text{Sensitive Sim}(a^x, a^y) = \frac{1}{2} \left( \frac{|\text{SC}_+(a^x) \cap \text{SC}_+(a^y)|}{|\text{SC}_+(a^x) \cup \text{SC}_+(a^y)|} + \frac{|\text{SC}_-(a^x) \cap \text{SC}_-(a^y)|}{|\text{SC}_-(a^x) \cup \text{SC}_-(a^y)|} \right)$$

(13)

Here we call the $\text{Sensitive Sim}(\cdot)$ as the sensitive Jaccard similarity.

After the sparse codeset for each data object is acquired, how to fast compute the Jaccard (or sensitive Jaccard) similarity given a sparse codeset for the query data object is significant. We provide two alternative strategies for fast retrieval on sparse codesets. Different from the hamming distance of binary hash codes, the jaccard similarity of sparse codesets should be taken a careful consideration as follows:

- Inspired by the fact that hamming distance can be calculated efficiently with bitwise operation [23], when the dictionary size $K$ is not large, each sparse codeset can be tiled as a $K$-dimensional binary vector with the nonzero elements indicating the indices selected coefficients. Since the intersection and union operation can be converted into bitwise AND and OR operations, the computation of Jaccard similarity of two sparse codesets is efficient.
- When the dictionary size $K$ is too large that the $K$-dimensional binary vector representation for all sparse codesets can not be loaded into memory all at once, an alternative strategy is using min-Hash [46] as the hash function for the sparse codesets and building LSH index [3] to provide efficient retrieval.

In our experiment, we adopt relatively small size dictionaries and thus the binary vector representation of all the sparse codesets can be stored in the memory for efficient bitwise Jaccard similarity computation. We will further reveal that even with a small size dictionary, our proposed SM$^2$H still has a superior performance.

IV. THE OPTIMIZATION OF SM$^2$H

In this section, we give a detailed optimization of SM$^2$H. (3) is a non-convex objective function with respect to $A$, $D^x$ and $D^y$, but it is convex with respect to $D^x$ and $D^y$ while fixing $A$ and vice versa. Therefore, in practice, we can develop an iterative algorithm to optimize $A$, $D^x$ and $D^y$ alternatively.

A. The Optimization of Sparse Coefficients

First, we fix $D^x$ and $D^y$ to optimize $A$. With $D^x$ and $D^y$ fixed, the optimization of $A$ can be obtained as follows:

$$\min_A \|X-D^x A^x\|^2_F + \|Y-D^y A^y\|^2_F + \|AA\|_1 + \alpha \text{Tr}(AL_k A^T)$$

(14)

Since the $\ell_1$-norm term is not differentiable at 0, we can’t directly use gradient descent method to solve (14). Sub-gradient descent strategy can be used, but such optimization has been observed to converge slowly in practice. Therefore, we adopt proximal method [47], which is much more efficient, to solve the problem. Denote the smooth part in (14) as:

$$f(A) = \|X-D^x A^x\|^2_F + \|Y-D^y A^y\|^2_F + \alpha \text{Tr}(AL_k A^T)$$

(15)

The optimization of updating sparse coefficients $A$ is described in Algorithm 1 and we adopt the implementation in MALSAR package [48].

B. The Optimization of Updating Multi-Modal Dictionaries

After obtaining $A$, we decompose $A$ to $A^x$ and $A^y$. Then the dictionaries $D^x$ and $D^y$ are updated as follows:

$$\min_{D^x, D^y} \|X-D^x A^x\|^2_F + \|Y-D^y A^y\|^2_F$$

subject to $\|d^x_k\|^2_F \leq 1, \|d^y_k\|^2_F \leq 1, \forall k = 1, 2, \ldots, K$
It is easy to figure out that $D^x$ and $D^y$ have not been coupled with each other in (16) so that we can update $D^x$ and $D^y$ independently as follows:

\begin{align}
\min_{D^x} & \| X - D^x A^x \|^2_F \\
\text{s.t.} & \| d^x_k \|^2_F \leq 1, \forall k = 1, 2, \ldots K \\
(17) \min_{D^y} & \| Y - D^y A^y \|^2_F \\
\text{s.t.} & \| d^y_k \|^2_F \leq 1, \forall k = 1, 2, \ldots K \label{eq:18}
\end{align}

(17) or (18) is a traditional least squares problem with quadratic constraints. We adopt Yang et al.’s method [49] to solve it.

We summarize the overall optimization of SM$^2$H in Algorithm 2.

---

**Algorithm 1** The optimization of updating sparse coefficients

**Input:** Initialized $A_0 = 0$, max iteration number $\tau$

**Output:** $A$

1: $A_1 = A_0$, $t = 0$, $t_0 = 1$
2: for $i = 1$ to $\tau$
3: $\mu_i = (t_{i-2} - 1)/t_{i-1}$, $S_i = A_i + \mu_i (A_i - A_{i-1})$
4: while true do
5: Compute $A^* = \arg\min P(A)$ where
6: $P(A) = f(S) + \| AA \|_1 + \langle \nabla f(S), A - S \rangle + \frac{\gamma}{2} \| A - S \|^2_F$
7: if $f(A^*) \leq P(A^*)$ then break the while loop
8: else $\gamma_i = \gamma_i \times 2$
9: end if
10: end while
11: $A_{i+1} = A^*$, $\gamma_{i+1} = \gamma_i$
12: if convergence then break the for loop
13: end if
14: $t_i = 1 + \sqrt{1 + 4t_{i-1}^2}/2$
15: end for
16: $A = A_{i+1}$

---

**Algorithm 2** The overall optimization of SM$^2$H

**Input:** $X$, $Y$, Initialized dictionaries $D^x$, $D^y$ with normalized columns

**Output:** The learned multi-modal dictionaries $D^x$ and $D^y$

1: Construct hypergraph $G(V, E, w^i)$ and compute hypergraph laplacian $L_h$
2: while not convergence do
3: solve (14) and get the joint sparse codes $A$ and then decompose it to $A^x$ and $A^y$
4: solve (16) to update the multi-modal dictionaries $D^x$ and $D^y$ based on the current $A^x$ and $A^y$
5: end while

---

**V. Complexity Analysis**

Since our method consists of an off-line multi-modal dictionary learning and on-line sparse codeset learning, we detail the time complexity for each part respectively.

**Off-Line Multi-Modal Dictionary Learning:** As aforementioned, the optimization of (3) is solved by an iterative manner. For each iteration, we solve the minimization problem of (14) and (16) successively. For problem (14), the time complexity is $O(p_x K + \tau K^2 N_x + K N_x^2)$. For problem (16), the time complexity is $O(p_y K N_y + \tau K N_y)$. Usually, it takes about 10–20 iterations for (3) to achieve convergence.

**On-Line Sparse Codeset Learning:** Given an out-of-sample data object $x$, its sparse coefficients is obtained by solving problem (8), which is a traditional lasso problem and can be solved in $O(p_x K)$ time. With some efficient multi-core optimized implementations such as SPAMS,$^1$ (8) can be solved within 10–40 second on a single machine. The conversion from sparse coefficients to sparse codeset can be finished in $O(K)$ time. Therefore, the time complexity for on-line sparse codeset learning is $O(p_x K)$.

---

**VI. Experiments and Comparisons**

In our experiments, we evaluate the performance of our proposed SM$^2$H. We first introduce the data set and evaluation criteria we adopted, then we elaborate parameter setting and tuning in our experiments. Finally, we compare our SM$^2$H with other state-of-the-art methods and demonstrate the results.

**A. Data Sets**

We use two real-world data sets “Wikipedia-Picture of the Day” (abbreviated as Wiki)$^2$ and NUS-WIDE.$^3$ Both data sets are bi-modal with images and texts.

The Wiki data set consists of 2866 Wikipedia documents which are provided by Rasiwasia et al. Each document contains a text and one corresponding relevant image. All documents are labeled by 10 semantic categories. For images, we extract SIFT descriptors [50] and then quantize them into Bag-of-Visual-Words (BoVW) by K-means clustering. For texts, we calculate the frequency of all words in the data set and select the most “representative” words (neither high-frequency nor low-frequency) to quantize all texts into Bag-of-Words (BoW).

As the dimensionality of data objects is a significant factor for cross media retrieval, we set two different kinds of dimensionality for comparisons: one is 500-D BoVW and 1,000-D BoW, the other is 1,000-D BoVW and 5,000-D BoW. We take 40% of the data objects as the training set and the remaining as the

---


$^2$http://www.svcl.ucsd.edu/projects/crossmodal/

$^3$http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
testing set. The query set is formed with random choosing 10% from the testing set.

The NUS-WIDE data set contains 186,577 labeled images. Each image is represented by 500-D BoVW and its corresponding tags is represented by a 1,000-D BoW. Additionally, the whole data set is manually annotated with 81 concepts. To be similar with [6], [15], [27], we choose $k$ largest concepts ($k = 10$ in our settings), each of which has abundant relevant images. Thereafter, we get about 70,000 images associated with their corresponding labels. We take 3% as the training set and the remaining as the testing set. The query set is formed with random choosing 1% from the testing set.

B. Evaluation Criteria

To evaluate our proposed SM$^2$H, we use a bi-directional retrieval policy. Specifically, we use image queries to retrieve texts (image-query-texts) and text queries to retrieve images (text-query-images) respectively.

To evaluate the performance of the cross-media retrieval results, three evaluation criteria are adopted, namely mean Average Precision (mAP), Percentage and Precision vs. Scope. They are defined as follows:

**mAP**: Following the strategy used in [15], we concentrate on the top $R$ results. Given a query and its top $R$ retrieved results, Average Precision is defined as:

$$AP = \frac{1}{L} \sum_{r=1}^{R} P(r) \delta(r)$$

where $L$ is the number of true neighbors of the query in the retrieved list. Here, for image-query-texts or text-query-images, those data objects which belong to the same category with the query are regarded as the true neighbors. $P(r)$ denotes the precision of the top $R$ retrieved results and $\delta(r) = 1$ if the $r$-th result is a true neighbor and 0 otherwise. For all queries, we calculate their AP values and use the average value to obtain the mAP. The larger mAP value is, the better the retrieval performance is. In our experiments, we set $R = 50$.

**Percentage**: Since the data sets we use consist of paired correspondence image-text. That is to say, for a query image (or text), there is only one truly matched ground-truth text (or image). Apart from mAP which uses category labels as ground truth, we are also interested in the position of the ground-truth text/image in the ranked list. In general, one image (or text) is considered correctly retrieved if it appears in the first $t\%$ of the ranked list of its corresponding retrieved texts (or images) according to [51]. $t$ is set to 0.2 in our experiments.

**Precision vs. Scope**: Additionally, the precision within different retrieval scopes can be reported with curve graphs. This criteria is used to measure the performance of image-query images retrieval scheme.

C. Compared Methods

We compare our method with the three following counterparts of multi-modal hashing:

- **CMSSH**: To the best of our knowledge, CMSSH [13] is the first multi-modal hashing approach. CMSSH uses standard AdaBoost to learn hash functions sequentially and each hash function is obtained by solving an SVD problem. However, the hash functions learned by CMSSH only preserve the inter-modality similarity but ignore the intra-modality similarity.

- **CVH**: CVH [14] can be regarded as a multi-modal extension of Spectral Hashing. It formulates an objective function that takes both inter-similarity and intra-similarity into account and minimizes the distance of hash codes for similar data objects and maximizes the distance for dissimilar data objects.

- **MLBE**: MLBE [15] uses graphical model to describe the intra and inter relationship for multi-modal data objects and the hash codes are obtained by maximum a posteriori estimation.

Our SM$^2$H method can be referred as two approaches depending on different hashing schemes and distance metrics: SM$^2$H$_1$ in term of Hashing Scheme I and Jaccard similarity, SM$^2$H$_2$ in term of Hashing Scheme II and sensitive Jaccard similarity.

D. Parameters Tuning

For CMSSH, CVH and MLBE, the most important parameter is the code length $m$. If $m$ is too small, the hash collision is serious which is impractical for the ANN retrieval of large scale data set. Hence, the code length $m$ we choose is 16, 32, 48, 64. For MLBE, the rest of the parameters we follow Zhen et al.’s original settings [15].

For our SM$^2$H, the parameters we used are tuned via the validation set. When constructing hypergraph, the number of neighbors in intra-modality is set to 50. The weight $w_{ij}$ of the heterogeneous hyperedges is set to 100 which achieves the best performance in our SM$^2$H. During the multi-modal dictionary learning, the parameter $\alpha$ is set to 5. The threshold $\sigma$ for hashing scheme is set to 0.01.

The cardinality of each sparse codeset generated by SM$^2$H for each data object is not fixed. For a fair evaluation, we should guarantee the cardinality of sparse codesets is nearly equal to the length of binary hashing codes of the comparing counterparts. For example, denoting the hamming code length of compared counterparts as $m$, the number of possible hash entries is $2^m$ in the three counterparts, while in our SM$^2$H, the number of hash entries is the cardinality of combination set $C(K, s)$, where $K$ is the size of the dictionary and $s$ is the average number of nonzero elements in all sparse coefficients.

However, it is hard to choose the appropriate $s$ and $K$ since there are so many possible combinations. Thus, we define sparse degree $d = s/K$. When $d$ is fixed, it is easier for us to give reasonable values of $s$ and $K$. By fixing the parameter $\alpha$ of the hypergraph regularization term, we tune the parameters $\Lambda$ and dictionary codebook size $K$ to get the performance over different sparse degree $d$ and $K$. We report the average mAP variations (the average mAP scores of image-query-texts and text-query-images) under the settings of $d$ ranges from 0.1 to 0.5, and $K$ is 50,100,150. The results in Fig. 2 show that when sparse degree $d \leq 0.3$, SM$^2$H achieves overall the best performance. Therefore, $C(25, 5)$, $C(50, 10)$, $C(100, 10)$ and $C(150, 15)$ are used in SM$^2$H, which correspond to the length of binary codes in comparing counterparts, namely, $m = 16$, $m = 32$, $m = 48$, $m = 64$, respectively.
Fig. 2. The obtained mAP values by different sparse degrees and dictionary sizes. It can be observed that the best performance is achieved for each $K$ with the sparse degree $\leq 0.3$.

**TABLE II**

<table>
<thead>
<tr>
<th>Task</th>
<th>Methods</th>
<th>code lengths or equivalent sizes of sparse codsets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m = 16$ or $C(25,5)$</td>
<td>$m = 32$ or $C(50,10)$</td>
</tr>
<tr>
<td>image-query-texts</td>
<td>CMSSH</td>
<td>0.4410</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.3675</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.3989</td>
</tr>
<tr>
<td></td>
<td>SM$^2$H$_1$</td>
<td>0.4279</td>
</tr>
<tr>
<td></td>
<td>SM$^4$H$_2$</td>
<td><strong>0.4496</strong></td>
</tr>
<tr>
<td>text-query-images</td>
<td>CMSSH</td>
<td>0.4013</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.3771</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.3989</td>
</tr>
<tr>
<td></td>
<td>SM$^2$H$_1$</td>
<td><strong>0.4689</strong></td>
</tr>
<tr>
<td></td>
<td>SM$^4$H$_2$</td>
<td>0.4577</td>
</tr>
</tbody>
</table>

**TABLE III**

<table>
<thead>
<tr>
<th>Task</th>
<th>Methods</th>
<th>code lengths or equivalent sizes of sparse codsets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m = 16$ or $C(25,5)$</td>
<td>$m = 32$ or $C(50,10)$</td>
</tr>
<tr>
<td>image-query-texts</td>
<td>CMSSH</td>
<td>0.3392</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.3050</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.2874</td>
</tr>
<tr>
<td></td>
<td>SM$^2$H$_1$</td>
<td>0.3912</td>
</tr>
<tr>
<td></td>
<td>SM$^4$H$_2$</td>
<td><strong>0.4119</strong></td>
</tr>
<tr>
<td>text-query-images</td>
<td>CMSSH</td>
<td>0.3011</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.3050</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.2696</td>
</tr>
<tr>
<td></td>
<td>SM$^2$H$_1$</td>
<td><strong>0.4470</strong></td>
</tr>
<tr>
<td></td>
<td>SM$^4$H$_2$</td>
<td><strong>0.4912</strong></td>
</tr>
</tbody>
</table>

$m = 32$, $m = 48$ and $m = 64$. We give detailed analysis on sparse degree in Section VI-E.

**E. Discussion on Sparse Degree**

From the results demonstrated in Fig. 2, we can make the following observations in our SM$^2$H: 1) the size of dictionary $K$ is not a key factor to influence the performance of SM$^2$H. With the increasing of $K$, the values of mAP scores do not change dramatically; 2) sparse degree has some effects on the performance of SM$^2$H. When the sparse degree $\leq 0.3$, the performance of SM$^2$H is relatively good, whereas the sparse degree is larger than 0.3, the performance of SM$^2$H decreases seriously.

A possible reason for observation 1 may be that the performance of our proposed SM$^2$H is heavily controlled by whether the informative component indices are activated from the sparse coefficients. For observation 2, as the sparse degree increases, some insignificant component indices are activated and some “noise” is introduced into the sparse codsets which consequently leads to a disappointed result shown in Fig. 2.
TABLE IV
THE PERFORMANCE COMPARISON IN TERMS OF mAP SCORES ON WIKI DATA SET WITH CODE LENGTH $m$ EQUALS TO 16, 32, 48 AND 64, AND THE EQUIVALENT SIZE OF SPARSE CODESETS IN SM$^2$H, 500-D BAG-OF-VISUAL-WORDS (BOVW) AND 1,000-D BAG-OF-TEXTUAL-WORDS (BOW), AS WELL AS 1,000-D BAG-OF-VISUAL-WORDS (BOVW) AND 5,000-D BAG-OF-TEXTUAL-WORDS (BOW), ARE USED TO REPRESENT EACH IMAGE AND TEXT RESPECTIVELY.
The items shown in bold are the two best results, the results with asterisk are the best.

<table>
<thead>
<tr>
<th>Task</th>
<th>Methods</th>
<th>BoVW(500D), BoW(1000D)</th>
<th>BoVW(1000D), BoW(5000D)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>code lengths or equivalent sizes of sparse code set</td>
<td>code lengths or equivalent sizes of sparse code sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 16$ or C(25,5)</td>
<td>$m = 16$ or C(25,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 32$ or C(50,10)</td>
<td>$m = 32$ or C(50,10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 48$ or C(100,10)</td>
<td>$m = 48$ or C(100,10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 64$ or C(150,15)</td>
<td>$m = 64$ or C(150,15)</td>
</tr>
<tr>
<td>image-query-texts</td>
<td>CMSSH</td>
<td>0.1097</td>
<td>0.1095*</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.1878</td>
<td>0.1850</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.1737</td>
<td>0.1709</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H</td>
<td>0.1912</td>
<td>0.2029</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H$_2$</td>
<td>0.1937*</td>
<td>0.2036</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1847</td>
<td>0.2038*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2254</td>
<td>0.2273*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2113*</td>
<td>0.2105*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1905</td>
<td>0.1964</td>
</tr>
<tr>
<td>text-query-images</td>
<td>CMSSH</td>
<td>0.1985*</td>
<td>0.2003</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.2021</td>
<td>0.1768</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.1819</td>
<td>0.1766</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H</td>
<td>0.2102</td>
<td>0.2082</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H$_2$</td>
<td>0.2139*</td>
<td>0.2156*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2205</td>
<td>0.2239*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2200</td>
<td>0.2231*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2260</td>
<td>0.2311*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2370</td>
<td>0.2336*</td>
</tr>
</tbody>
</table>

TABLE V
THE PERFORMANCE COMPARISON IN TERMS OF PERCENTAGE SCORES ON WIKI DATA SET WITH CODE LENGTH $m$ EQUALS TO 16, 32, 48 AND 64, AND THE EQUIVALENT SIZE OF SPARSE CODESETS IN SM$^2$H, 500-D BAG-OF-VISUAL-WORDS (BOVW) AND 1,000-D BAG-OF-TEXTUAL-WORDS (BOW), AS WELL AS 1,000-D BAG-OF-VISUAL-WORDS (BOVW) AND 5,000-D BAG-OF-TEXTUAL-WORDS (BOW), ARE USED TO REPRESENT EACH IMAGE AND TEXT RESPECTIVELY.
The items shown in bold are the two best results, the results with asterisk are the best.

<table>
<thead>
<tr>
<th>Task</th>
<th>Methods</th>
<th>BoVW(500D), BoW(1000D)</th>
<th>BoVW(1000D), BoW(5000D)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>code lengths or equivalent sizes of sparse code set</td>
<td>code lengths or equivalent sizes of sparse code sets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 16$ or C(25,5)</td>
<td>$m = 16$ or C(25,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 32$ or C(50,10)</td>
<td>$m = 32$ or C(50,10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 48$ or C(100,10)</td>
<td>$m = 48$ or C(100,10)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$m = 64$ or C(150,15)</td>
<td>$m = 64$ or C(150,15)</td>
</tr>
<tr>
<td>image-query-texts</td>
<td>CMSSH</td>
<td>0.2544</td>
<td>0.2450</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.2509</td>
<td>0.2533</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.2827</td>
<td>0.2903</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H</td>
<td>0.3225</td>
<td>0.3216</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H$_2$</td>
<td>0.3512*</td>
<td>0.3534*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3816*</td>
<td>0.3922*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3640*</td>
<td>0.3498*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3074</td>
<td>0.4099*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3110</td>
<td>0.3993*</td>
</tr>
<tr>
<td>text-query-images</td>
<td>CMSSH</td>
<td>0.2512</td>
<td>0.2580</td>
</tr>
<tr>
<td></td>
<td>CVH</td>
<td>0.2509</td>
<td>0.2586</td>
</tr>
<tr>
<td></td>
<td>MLBE</td>
<td>0.2827</td>
<td>0.2403</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H</td>
<td>0.3089</td>
<td>0.3151</td>
</tr>
<tr>
<td></td>
<td>SM$^3$H$_2$</td>
<td>0.3183*</td>
<td>0.3392*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.4064*</td>
<td>0.4417*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.3710</td>
<td>0.4208*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.4170*</td>
<td>0.4170*</td>
</tr>
</tbody>
</table>

Fig. 3. The performance comparison of image-query-images in terms of Precision vs. Scope curve with code length $m = 32$ for CVH, CMSSH, MLBE and $C(50, 10)$ for SM$^3$H$_1$ and SM$^3$H$_2$.

F. Performance Comparisons
For the NUS-WIDE data set, the performance is given in Table II and Table III. For the Wiki data set, the performance is given in Table IV and Table V. Besides, we conduct additional experiments of image-query-images retrieval. The precision vs. scope curves demonstrated in Fig. 3 is the comparison with code length $m = 32$ or equivalent size of sparse code sets $C(50, 10)$. In image-query-images retrieval, the data objects belong to the same category with the query image are regarded as the true neighbors.
From the experiments, we can make the following observations:
Fig. 4. Some examples of image-query-texts direction (in top half) and text-query-images direction (in bottom half) on the Wiki data set. For each direction, we show the query and its corresponding top retrieved results by our proposed $\text{SM}^2\text{H}$ and three counterparts. The incorrect retrieved results are in red frame. For the image-query-texts direction, we can find that given a “battleship” image belongs to the “warfare” category, the top retrieved texts of our $\text{SM}^2\text{H}$ are clearly relevant while the three counterparts all produce some irrelevant results; for the text-query-images direction, given a text about “church”, “building”, etc, the top retrieved images of $\text{SM}^2\text{H}$ are the most relevant compared with the counterparts. Moreover, the retrieved result of $\text{SM}^2\text{H}$ is more accurate than $\text{SM}^2\text{H}_1$ in human understanding.

- The proposed $\text{SM}^2\text{H}$ achieves the overall best performance in terms of mAP and Percentage over the Wiki dataset and the NUS-WIDE dataset, due to its appealing nature of the modeling of multi-modal correlations (intra-modality and inter-modality) and the identification of sparse meaningful component indices.
- In most cases, the performance of three counterparts will degrade when the code length increases. For example, MLBE may get trapped in local minima during the learning process when the code length is too large [15]; the projection directions of CVH is based on the eigen vectors of the minimal eigenvalues, thus hash codes with a longer length may lead to unfaithful projection directions; MLBE does not require the independency between different hash bits, and thus may generate highly redundant bits. However, the performance of our proposed $\text{SM}^2\text{H}$ is fundamentally controlled by whether the informative component indices are activated or not, and
the cardinality of sparse codesets seems to influence the performance of SM²H insignificantly. From a feature selection point of view, SM²H sets up more interpretable model for multi-modal hashing.

- SM²H only slightly outperforms SM²H in terms of mAP over the two data sets. This phenomenon may be caused by the category based ground truth we used since the categories (e.g., ‘art’, ‘history’ or ‘biology’ in Wiki data set) are somehow too general to exactly describe the detailed semantics of multi-modal data objects. On the contrary, the Percentage criteria can be seen a more accurate indicator of true performance, since images and texts are paired in our experiments. As we expected, SM²H outperforms SM²H significantly in terms of Percentage over both of the data sets.

- Not only does our SM²H method have superior performance on cross-media retrieval, the image-query-images results in Fig. 3 also demonstrate that our SM²H dramatically outperforms the three counterparts on preserving the intra-modality similarity. Moreover, SM²H outperforms SM²H owing to its more accurate representation.

Finally, we demonstrate some examples of retrieved results on two retrieval directions (image-query-texts and text-query-images) in Fig. 4. Based on intuitive judgement, we can get the observation that SM²H achieves the best results comparing with three counterparts. Furthermore, SM²H outperforms SM²H which proves the effectiveness of sensitive Jaccard metric on sparse codesets.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a sparse multi-modal hashing (SM²H) approach. SM²H can activate those significantly informative component indices of sparse coefficients to generate a sparse codeset for each data object. To measure the similarity of two sparse codesets (hash codes), a sensitive Jaccard similarity instead of the hamming distance is conducted for efficient ANN search.

Our proposed SM²H requires the tuning of \( \lambda_c \) and \( \lambda_y \) to have multi-modal data objects hold nearly equal level of sparsity. In our future work, a possible direction is to find a smarter way which can easily control these two parameters to make our proposed method even more competitive. Furthermore, the category of each multi-modal data object can also be exploited to establish the inter-modality relationships, i.e., the multi-modal data objects belonging to the same category can be connected by a high-order hyperedge. How to devise a more appropriate scheme to encode the inter-modality relationships poses a great challenge to further improving the performance of our SM²H method.

REFERENCES


Fei Wu received the B.S. degree from Lanzhou University, Lanzhou, Gansu, China, the M.S. degree from Macau University, Taipa, Macau, and the Ph.D. degree from Zhejiang University, Hangzhou, China. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. He was a Visiting Scholar with Prof. B. Yu’s Group, University of California, Berkeley, from 2009 to 2010. His current research interests include multimedia retrieval, sparse representation, and machine learning.

Zhou Yu received the B.S. degree from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2010, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. His current research interests include multimedia retrieval, machine learning, and large scale unstructured data management.

Yi Yang is a DECRA Fellow in the School of Information Technology and Electrical Engineering, The University of Queensland. Prior to that, he was a post-doctoral fellow with the School of Computer Science at CMU. He received his Ph.D. degree from Zhejiang University in Computer Science in 2010. His research interests include multimedia analysis, text mining, and statistic learning.

Siliang Tang received the B.S. degree from Zhejiang University, Hangzhou, Zhejiang, China, and the Ph.D. degree from National University of Ireland, Maynooth, Maynooth Co. Kildare, Ireland. He is currently a Lecturer of the College of Computer Science, Zhejiang University. His current research interests include multimedia analysis, text mining, and statistic learning.

Yin Zhang received his B.S. degree in Electrical Engineering and Ph.D. degree in Computer Science from Zhejiang University, Hangzhou, China, in 2002 and 2009 respectively. He is currently a faculty member of the College of Computer Science at Zhejiang University. His research interests lie in collaborative filtering, multimedia retrieval and their applications in digital libraries.

Yueting Zhuang received the B.S., M.S., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively. Currently, he is a Professor and Ph.D. Supervisor with the College of Computer Science, Zhejiang University. His current research interests include multimedia databases, artificial intelligence, and video-based animation.