SDMCH: Supervised Discrete Manifold-Embedded Cross-Modal Hashing

Xin Luo, Xiao-Ya Yin, Liqiang Nie, Xuemeng Song, Yongxin Wang, Xin-Shun Xu*
School of Computer Science and Technology, Shandong University
School of Software, Shandong University
{luoxin.lxin, xysyin, nieliqiang,sxmustc}@gmail.com, yxinwang@hotmail.com, xuxinshun@sdu.edu.cn

Abstract

Cross-modal hashing methods have attracted considerable attention. Most pioneer approaches only preserve the neighborhood relationship by constructing the correlations among heterogeneous modalities. However, they neglect the fact that the high-dimensional data often exists on a low-dimensional manifold embedded in the ambient space and the relative proximity between the neighbors is also important. Although some methods leverage the manifold learning to generate the hash codes, most of them fail to explicitly explore the discriminative information in the class labels and discard the binary constraints during optimization, generating large quantization errors. To address these issues, in this paper, we present a novel cross-modal hashing method, named Supervised Discrete Manifold-Embedded Cross-Modal Hashing (SDMCH). It can not only exploit the non-linear manifold structure of data and construct the correlation among heterogeneous multiple modalities, but also fully utilize the semantic information. Moreover, the hash codes can be generated discretely by an iterative optimization algorithm, which can avoid the large quantization errors. Extensive experimental results on three benchmark datasets demonstrate that SDMCH outperforms ten state-of-the-art cross-modal hashing methods.

1 Introduction

In the last several decades, cross-modal retrieval has attracted lots of attention as it has played a fundamental role in many fields such as visual search, machine translation, and text mining. Typically, in the cross-modal retrieval, one can use a query with one type of modality (e.g., text) to retrieve relevant items with another type of modality (e.g., image). More recently, to tackle the large-scale data, hashing based cross-modal retrieval methods have drawn more and more attention due to the fast query speed and low storage cost [Xu, 2016; Zhang *et al.*, 2017]. Roughly speaking, cross-modal hashing methods can be divided into unsupervised and supervised. The former learns the hash functions and hash codes by exploiting the inter- and intra-modality relatedness of the given data without utilizing the supervised information. The representative unsupervised hashing methods include Inter-Media Hashing (IMH) [Song *et al.*, 2013], Latent Semantic Sparse Hashing (LSSH) [Zhou *et al.*, 2014], Collective Matrix Factorization Hashing (CMFH) [Ding *et al.*, 2014], Composite Correlation Quantization (CCQ) [Long *et al.*, 2016], and Fusion Similarity Hashing (FSH) [Liu *et al.*, 2017]. The latter can further utilize the semantic information to learn the similarity-preserving hash codes and they have demonstrated better performance than that of unsupervised ones in many real-world applications. The representative supervised methods include Cross View Hashing (CVH) [Kumar and Udupa, 2011], Semantic Correlation Maximization (SCM) [Zhang and Li, 2014], Semantics Preserving Hashing (SePH) [Lin *et al.*, 2015], and Discrete Cross-Modal Hashing (DCH) [Xu *et al.*, 2017]. More recently, some deep cross-modal hashing models have been proposed and obtained promising performance [Jiang and Li, 2017; Yang *et al.*, 2017a].

As shown above, although many cross-modal hashing methods have been explored, there still remains several problems that need to be further explored. 1) Most of them fail to exploit the manifold structure which is important to boost the performance of a model. 2) Most of them only preserve the neighborhood relationship while neglect the relative neighborhood proximity. 3) Many hashing methods leverage the manifold learning to generate the binary codes; however, some fail to fully utilize the semantic information and some relax the binary constraints, generating large quantization errors.

To address the aforementioned challenges, in this paper, we present a novel cross-modal hashing method, named Supervised Discrete Manifold-Embedded Cross-Modal Hashing (SDMCH). It can not only exploit the non-linear manifold structure of data, but also construct the correlations among heterogeneous multiple modalities. In addition, it fully utilizes the semantic information to preserve the similarity and generate more precise hash codes. Based on an iterative algorithm proposed in this work, instead of relaxing the binary constraints, SDMCH generates the hash codes directly. Therefore, SDMCH can avoid the large quantization errors. Moreover, SDMCH is a two-step hashing method, making it flexible and effective. Extensive experimental results on three

*Corrsponding Author.

widely used benchmark datasets demonstrate its superiority over several state-of-the-art cross-modal hashing methods.

2 Related Work

2.1 Hashing with Manifold Learning

Recent studies have demonstrated that it is beneficial for a retrieval model to exploit the non-linear manifold structure of multimedia data [Yang *et al.*, 2008; Zhang *et al.*, 2015]. Therefore, many methods leverage the manifold learning to generate the hash codes. For example, one well-known work is Spectral Hashing [Weiss *et al.*, 2009], generating the hash codes by thresholding the Laplace-Beltrami eigenfunctions of the manifolds. Some other representative manifold hashing methods include Anchor Graph Hashing [Liu *et al.*, 2011], Locally Linear Hashing [Irie *et al.*, 2014], Inductive Manifold Hashing [Shen *et al.*, 2015], and Discrete Locally Linear Embedding Hashing [Ji *et al.*, 2017]. However, these methods can only work on unimodal data. To deal with the data with multiple modalities, some manifold based multimodal hashing methods have been explored, including Composite Hashing with Multiple Information Sources [Zhang *et al.*, 2011], Multiple Feature Hashing [Song *et al.*, 2011], Multi-View Complementary Hash [Liu *et al.*, 2015], and Discrete Multi-View Hashing [Yang *et al.*, 2017b]. However, these models cannot perform cross-modal retrieval, limiting their applications.

More recently, although some cross-modal hashing methods have been explored to leverage the manifold learning, there are still several issues that need to be further considered. For example, some cannot fully utilize the semantic information, e.g., Inter-Media Hashing [Song *et al.*, 2013], Sparse Multi-Modal Hashing [Wu *et al.*, 2014], and Full-Space Local Topology Extraction [Zhang *et al.*, 2015]. Some relax the binary constraints during optimization, generating large quantization errors, e.g., Cross-View Hashing [Kumar and Udupa, 2011], Multimodal Similarity-Preserving Hashing [Masci *et al.*, 2014], Supervised Matrix Factorization Hashing [Tang *et al.*, 2016], and Hetero-Manifold Regularisation [Zheng *et al.*, 2017].

2.2 Two-Step Hashing

Two-step hashing [Lin *et al.*, 2013; 2015; Luo *et al.*, 2018] decomposes the hash learning problem into two steps: 1) The binary code inference step; 2) the hash function learning step. And it has attracted broad research interests due to its simplicity, flexibility and effectiveness.

In the second step, given the learnt hash codes, a two-step hashing method learns hash functions to transform the original features into the compact binary codes. Actually, as Lin et al. revealed [Lin *et al.*, 2013], for any bit of the hash code, learning the corresponding hash function to project features into it can be modelled as a binary classification problem and the learning of hash function is open for any effective predictive model, like SVM, boosted decision trees and deep networks. Therefore, the main differences among the two-step hashing methods lie in the first step, which determines the quality of the generated binary codes. Consequently, the two-step hashing methods focus more on how to design effective loss functions and optimization algorithms.

3 Our Method

3.1 Notations

Suppose there are n training instances and m different modalities. $\mathbf{X}^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_n^t] \in \mathbb{R}^{n \times d_t}$ is the feature matrix of the t-th modality, where d_t denotes the dimensionality of its feature space and $t = 1, \cdots, m$. Without loss of generality, we further suppose that the instances are zero-centered in each modality, i.e., $\sum_{i=1}^{n} \mathbf{x}_i^t = 0$. In addition, the label matrix for all training instances are $\mathbf{Y} \in \{0, 1\}^{n \times c}$, where c is the number of classes, $Y_{ik} = 1$ denotes that $\mathbf{x}_i$ belongs to class k and 0 otherwise. $\mathbf{B} \in \{-1, 1\}^{n \times r}$ is the to-be-learnt hash code matrix for the training data, r is the length of the hash codes. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix and $Tr(\cdot)$ is the trace of a matrix. $sgn(\cdot)$ is an element-wise sign function defined as $sgn(x) = 1$ if $x \geq 0$, and -1 otherwise.

3.2 Manifold Structure Learning

To embed the manifold structure into the binary codes learning, we first construct the similarity matrix by leveraging Locally Linear Embedding [Roweis and Saul, 2000]. Specifically, one point is approximated as a weighted linear combination of its *K*-nearest neighbours. The formulation is defined as follows,

$$\min_{\mathbf{P}^t} \sum_{i=1}^{n} \|\mathbf{x}_i^t - \sum_{j}^{K} \mathbf{P}_{ij}^t \mathbf{x}_j^t\|_F^2, \;\; s.t. \sum_{j} \mathbf{P}_{ij}^t = 1, \qquad (1)$$

where $\mathbf{P}_{ij}^t = 0$ if x_i and x_j are not neighbours in the t-th modality and K is the number of neighbors of one instance. By solving Eq. (1), we can learn the weights $\mathbf{P}^t$ $(t = 1, \cdots, m)$. To save space, in this work, we do not give the details of how to solve the above problem, which can be found in [Roweis and Saul, 2000]. Besides, other manifold approaches can also be used to substitute LLE in our model. As it is not the focus of this paper to study different manifold methods, we just use LLE for illustrating the effectiveness of SDMCH.

Thereafter, the manifold structure preserving similarity matrix $\mathbf{S}^t$ is defined as $\mathbf{S}_{ij}^t = |\mathbf{P}_{ip}^t|$ if $\mathbf{x}_j^t$ is the p-th neighbor of $\mathbf{x}_i^t$ and $\mathbf{S}_{ij}^t = 0$ otherwise, where $\mathbf{P}_{ip}^t$ is the j-th element of $\mathbf{P}_i^t$. In addition, to ensure the symmetry of $\mathbf{S}^t$, we further update it with $\mathbf{S}^t \leftarrow ((\mathbf{S}^t)^\top + \mathbf{S}^t)$.

3.3 Objective Function of SDMCH

Manifold Embedding. To preserve the manifold similarity in the Hamming space, we further define the following problem,

$$\begin{aligned} \min_{\mathbf{F},\mathbf{B}} \quad & \sum_{t=1}^{m} \lambda_t \sum_{i,j=1}^{n} \mathbf{S}_{ij}^t \parallel \mathbf{F}_i - \mathbf{F}_j \parallel_F^2 + \alpha \parallel \mathbf{B} - \mathbf{F} \parallel_F^2, \\ & = \sum_{t=1}^{M} \lambda_t Tr(\mathbf{F}^\top \mathbf{L}^t \mathbf{F}) + \alpha \parallel \mathbf{B} - \mathbf{F} \parallel_F^2, \\ s.t. \quad & \mathbf{B} \in \{-1, 1\}^{n \times r}, \end{aligned} \qquad (2)$$

where $\mathbf{L}^t = \mathbf{D}^t - \mathbf{S}^t$, and $\mathbf{D}^t$ is a diagonal matrix whose entries are given by $\mathbf{D}^t_{ii} = \sum_{j=1}^{n} \mathbf{S}^t_{ij}$. In addition, $\mathbf{F}$ is a temporary real-valued representation used to approximate $\mathbf{B}$. Therefore, the first term is used to preserve the manifold similarity in the temporary representation space; the second one is to make $\mathbf{F}$ approximate $\mathbf{B}$ as much as possible.

Label Preserving. To make full use of the label information, we further assume that the labels can be obtained from the binary codes. For this purpose, we define the following formulation,

$$\min_{G} \quad \| \mathbf{Y} - \mathbf{BG} \|_F^2, \tag{3}$$

where $\mathbf{G}$ is a projection matrix, mapping $\mathbf{B}$ to $\mathbf{Y}$.

Overall Objective Function. Combining Eq. (2) and (3), we obtain the overall objective function of SDMCH,

$$\begin{aligned} \min_{\mathbf{F},\mathbf{Z},\mathbf{G},\mathbf{B}} \quad & \sum_{t=1}^{m} \lambda_t Tr(\mathbf{F}^\top \mathbf{L}^t \mathbf{F}) + \alpha \| \mathbf{B} - \mathbf{F} \|_F^2 \\ & + \beta \| \mathbf{B} - \mathbf{Z} \|_F^2 + \gamma \| \mathbf{Y} - \mathbf{ZG} \|_F^2, \\ s.t. \quad & \mathbf{B} \in \{-1, 1\}^{n \times r}, \end{aligned} \tag{4}$$

where $\lambda_t, \alpha, \beta, \gamma$ are balance parameters. It is worth noting that, in the fourth term, we replace $\mathbf{B}$ with a semantic-preserving latent representation $\mathbf{Z}$. In addition, the third term is introduced to make $\mathbf{Z}$ approximate $\mathbf{B}$ as much as possible. Intuitively, by solving Eq. (4), both the manifold structure and semantic information can be embedded into the hash codes. Moreover, due to the use of $\mathbf{F}$ and $\mathbf{Z}$, the problem becomes easy to be efficiently solved by an iterative optimization scheme without relaxation as shown in the following subsection.

3.4 Optimization

To solve the optimization problem in Eq. (4), we propose a four-step iterative scheme as shown below.

F Step. When $\mathbf{Z}, \mathbf{G}$, and $\mathbf{B}$ are fixed, the optimization problem can be formulated as,

$$\min_{\mathbf{F}} \sum_{t=1}^{m} \lambda_t Tr(\mathbf{F}^\top \mathbf{L}^t \mathbf{F}) + \alpha \| \mathbf{B} - \mathbf{F} \|_F^2 . \tag{5}$$

Setting the derivative of Eq. (5) w.r.t. $\mathbf{F}$ to zero, we have,

$$\mathbf{F} = \mathbf{AB}, \tag{6}$$

where $\mathbf{A} = \alpha(\sum_{t=1}^{m} \lambda_t \mathbf{L}^t + \alpha \mathbf{I})^{-1}$ is constant and can be precomputed.

Z Step. Fixing $\mathbf{F}, \mathbf{G}$, and $\mathbf{B}$, the problem in (4) is transformed to,

$$\min_{\mathbf{Z}} \beta \| \mathbf{B} - \mathbf{Z} \|_F^2 + \gamma \| \mathbf{Y} - \mathbf{ZG} \|_F^2 . \tag{7}$$

Setting the derivative of Eq. (7) w.r.t. $\mathbf{Z}$ to zero, we have,

$$\mathbf{Z} = (\beta \mathbf{B} + \gamma \mathbf{YG}^\top)(\beta \mathbf{I} + \gamma \mathbf{GG}^\top)^{-1}. \tag{8}$$

G Step. When $\mathbf{F}, \mathbf{Z}$, and $\mathbf{B}$ are fixed, Eq. (4) becomes,

$$\min_{\mathbf{G}} \| \mathbf{Y} - \mathbf{ZG} \|_F^2 . \tag{9}$$

Algorithm 1 Optimization algorithm in SDMCH.

Input: The manifold-embedded similarity matrices $\mathbf{S}^t$, the label matrix $\mathbf{Y}$, the parameters λ_t, α, β and γ, the hash code length r, and the iterative number T.

```
Randomly initialize F, Z, G, and B.
for iter = 1 → T do
    Fix Z, G, and B, and update F using Eq (6).
    Fix F, G, and B, and update Z using Eq (8).
    Fix F, Z, and B, and update G using Eq (10).
    Fix F, Z, and G, and update B using Eq (12).
end for
```

Output: $\mathbf{F}, \mathbf{Z}, \mathbf{G}$, and $\mathbf{B}$.

It is a least squares problem w.r.t $\mathbf{G}$, which has a closed-form solution,

$$\mathbf{G} = (\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{Y}. \tag{10}$$

B Step. When $\mathbf{F}, \mathbf{Z}$, and $\mathbf{G}$ are fixed, we can transform all terms into the trace operations, omit the constant terms, and rewrite Eq. (4) as follows,

$$\begin{aligned} \max_{\mathbf{B}} \quad & \alpha Tr(\mathbf{B}^\top \mathbf{F}) + \beta Tr(\mathbf{B}^\top \mathbf{Z}), \\ s.t. \quad & \mathbf{B} \in \{-1, 1\}^{n \times r}. \end{aligned} \tag{11}$$

Thereafter, the optimal solution can be obtained,

$$\mathbf{B} = sgn(\alpha \mathbf{F} + \beta \mathbf{Z}). \tag{12}$$

The Whole Algorithm. The above optimization procedure is summarized in Algorithm 1.

3.5 Hash Functions Learning

As mentioned previously, SDMCH is a two-step hashing method. In the second step, it adopts the linear regression as the hash functions. Specifically, given the binary codes $\mathbf{B}$ learnt in the first step, the hash projection matrix for the t-th modality, i.e., $\mathbf{W}^t$, is learnt by optimizing the following problem,

$$\| \mathbf{B} - \mathbf{X}^t \mathbf{W}^t \|_F^2 + \theta \| \mathbf{W}^t \|_F^2, \tag{13}$$

where θ is a balance parameter, and $\| \mathbf{W}^t \|_F^2$ is a regularization term. Thereafter, the optimal $\mathbf{W}^t$ is computed with the following equation,

$$\mathbf{W}^t = ({\mathbf{X}^t}^\top \mathbf{X}^t + \theta \mathbf{I})^{-1} {\mathbf{X}^t}^\top \mathbf{B}. \tag{14}$$

Once we have obtained $\mathbf{W}^t$, for the query data with the t-th modality $\mathbf{X}^t_{query}$, the hash codes are generated as follows,

$$\mathbf{B}_{query} = sgn(\mathbf{X}^t_{query} \mathbf{W}^t). \tag{15}$$

4 Experiments

To evaluate the performance of SDMCH, we conducted extensive experiments on three benchmark datasets and compared it with ten state-of-the-art hashing methods.

Task	Method	Wiki				MIRFlickr				NUS-WIDE			
		32bits	64bits	96bits	128bits	32bits	64bits	96bits	128bits	32bits	64bits	96bits	128bits
Image -to- Text	CCQ	0.1680	0.1682	0.1685	0.1683	0.5746	0.5751	0.5740	0.5749	0.3877	0.3886	0.3885	0.3886
	CMFH	0.2256	0.2345	0.2352	0.2440	0.5719	0.5718	0.5731	0.5729	0.3477	0.3416	0.3412	0.3370
	CVH	0.1483	0.1502	0.1488	0.1488	0.5699	0.5673	0.5657	0.5648	0.3557	0.3502	0.3479	0.3467
	DCH	0.3589	0.3777	0.3797	0.3791	0.6847	0.6782	0.6864	0.6938	0.5985	0.5886	0.5759	0.5775
	FSH	0.2386	0.2566	0.2514	0.2653	0.5922	0.6010	0.6040	0.6044	0.4371	0.4534	0.4613	0.4680
	IMH	0.1637	0.1655	0.1684	0.1688	0.5686	0.5677	0.5660	0.5659	0.3504	0.3509	0.3495	0.3479
	LSSH	0.2109	0.2095	0.2102	0.2024	0.5708	0.5757	0.5772	0.5786	0.3879	0.3927	0.3913	0.3912
	SCM-orth	0.1387	0.1286	0.1309	0.1286	0.5765	0.5723	0.5676	0.5689	0.3730	0.3614	0.3545	0.3528
	SCM-seq	0.2410	0.2437	0.2501	0.2541	0.6357	0.6473	0.6504	0.6526	0.5311	0.5330	0.5349	0.5400
	SePH-km	0.2820	0.3076	0.3030	0.3137	0.6873	0.6882	0.6889	0.6874	0.5440	0.5449	0.5760	0.5510
	SDMCH	**0.3843**	**0.3924**	**0.4011**	**0.4025**	**0.7089**	**0.7210**	**0.7258**	**0.7262**	**0.6235**	**0.6393**	**0.6414**	**0.6437**
Text -to- Image	CCQ	0.2518	0.2508	0.2543	0.2542	0.5986	0.5996	0.5989	0.5998	0.5986	0.5996	0.5989	0.5998
	CMFH	0.5295	0.5316	0.5392	0.5430	0.5665	0.5665	0.5670	0.5663	0.3505	0.3437	0.3441	0.3400
	CVH	0.1398	0.1293	0.1279	0.1259	0.5699	0.5674	0.5663	0.5658	0.3552	0.3509	0.3489	0.3479
	DCH	0.7097	0.7216	0.7212	0.7141	0.7601	0.7665	0.7792	0.7882	0.7303	0.7162	0.7065	0.6914
	FSH	0.5037	0.5228	0.5188	0.5329	0.5958	0.6056	0.6100	0.6168	0.4522	0.4696	0.4836	0.4916
	IMH	0.1400	0.1324	0.1326	0.1320	0.5687	0.5672	0.5666	0.5665	0.3484	0.3483	0.3479	0.3464
	LSSH	0.5220	0.5284	0.5301	0.5332	0.5894	0.5923	0.5928	0.5929	0.4211	0.4221	0.4184	0.4162
	SCM-orth	0.1338	0.1277	0.1255	0.1219	0.5736	0.5670	0.5656	0.5645	0.3702	0.3591	0.3513	0.3531
	SCM-seq	0.2459	0.2480	0.2469	0.2530	0.6214	0.6285	0.6337	0.6358	0.5101	0.5159	0.5137	0.5191
	SePH-km	0.6451	0.6662	0.6669	0.6706	0.7456	0.7476	0.7492	0.7497	0.6358	0.6405	0.6774	0.6391
	SDMCH	**0.7670**	**0.7701**	**0.7675**	**0.7709**	**0.7832**	**0.8102**	**0.8171**	**0.8169**	**0.7462**	**0.7659**	**0.7715**	**0.7706**

Table 1: The MAP results of all methods on three datasets. The best MAPs for each category are shown in boldface.

4.1 Datasets

Experiments are conducted on three benchmark datasets, i.e., Wiki [Rasiwasia *et al.*, 2010], MIRFlickr [Huiskes and Lew, 2008], and NUS-WIDE [Chua *et al.*, 2009], which are described below.

Wiki consists of $2,866$ image-text pairs collected from Wikipedia. Each instance is annotated with one of 10 semantic classes. The image and text of each instance are represented by a 128-dimension SIFT feature vector and a 10-dimension topic vector, respectively. It was initially split into a training set with $2,173$ instances and a test set of 693 instances.

MIRFlickr consists of $25,000$ instances collected from Flickr website, each being manually annotated with at least one of 24 labels. The image and text modalities are represented by a 150-dimension edge histogram and a 500-dimension vector generated from PCA on its index tagging vector, respectively. We randomly selected 25% instances of the dataset as the query set, and the remaining 75% ones as the training set.

NUS-WIDE contains $269,648$ instances crawled from Flickr. Each instance has an image with its associated textual tags and is manually annotated with at least one of 81 labels. However, some instances' labels are scare; so we followed the same setting of previous works [Zhou *et al.*, 2014; Lin *et al.*, 2015; Xu *et al.*, 2017] and selected 10 most common concepts and the corresponding $186,577$ instances. For each instance, the visual view is represented by a 500-dimension bag-of-visual SIFT feature vector and the textual view is represented by a binary tagging vector w.r.t. the top $1,000$ most frequent tags. We select 2% of the data as the query set and the rest as the training set.

Due to the large computational cost of some baselines, we randomly selected 5,000 instances on MIRFlickr and 10,000 instances on NUS-WIDE from the original training set to train all models.

4.2 Baselines and Implementation Details

Considering that SDMCH is not a deep model, we compared SDMCH with ten shallow state-of-the-art cross-modal hashing methods including: five unsupervised ones, i.e., IMH, LSSH, CMFH, CCQ, and FSH; five supervised ones, i.e., CVH, SCM-orth, SCM-seq, SePH-km, and DCH. The source codes of most baselines are kindly provided by the authors. We carefully tuned their parameters according to the scheme suggested by the authors. All experiments are repeated several times with random data partitions, and the averaged results are reported.

For SDMCH, its parameters are set to $K = 6$, $\lambda_1 = 0.3$, $\lambda_2 = 0.7$, $\alpha = 0.3$, $\beta = 5$, $\gamma = 1000$ and $\theta = 1$, selected by a validation procedure. In addition, the total iterative number T in Algorithm 1 is set to 10.

4.3 Evaluation Metrics

In the experiments, we conducted two cross-modal retrieval tasks: 1) Image-to-Text using images as the query to search texts; 2) Text-to-Image using texts as the query to search images. We adopted three widely-used performance metrics for multi-modal hashing to evaluate all methods, namely mean average precision (MAP), top-N precision curves, and precision-recall curves.

4.4 Overall Comparison with Baselines

MAP Results: The MAP values of SDMCH and all baselines on the three datasets are summarized in Table 1. From this table, we have the following observations. First, SDMCH obtains the best results in all cases and performs much better than the baselines in some cases, well demonstrating its effectiveness on the datasets. Second, all of the methods generally perform better at the Text-to-Image task than Image-to-Text. The main reason is that the text modality can better describe the content of an image-text pair than the image modality. Third, supervised methods, such as DCH, SePH-km, and

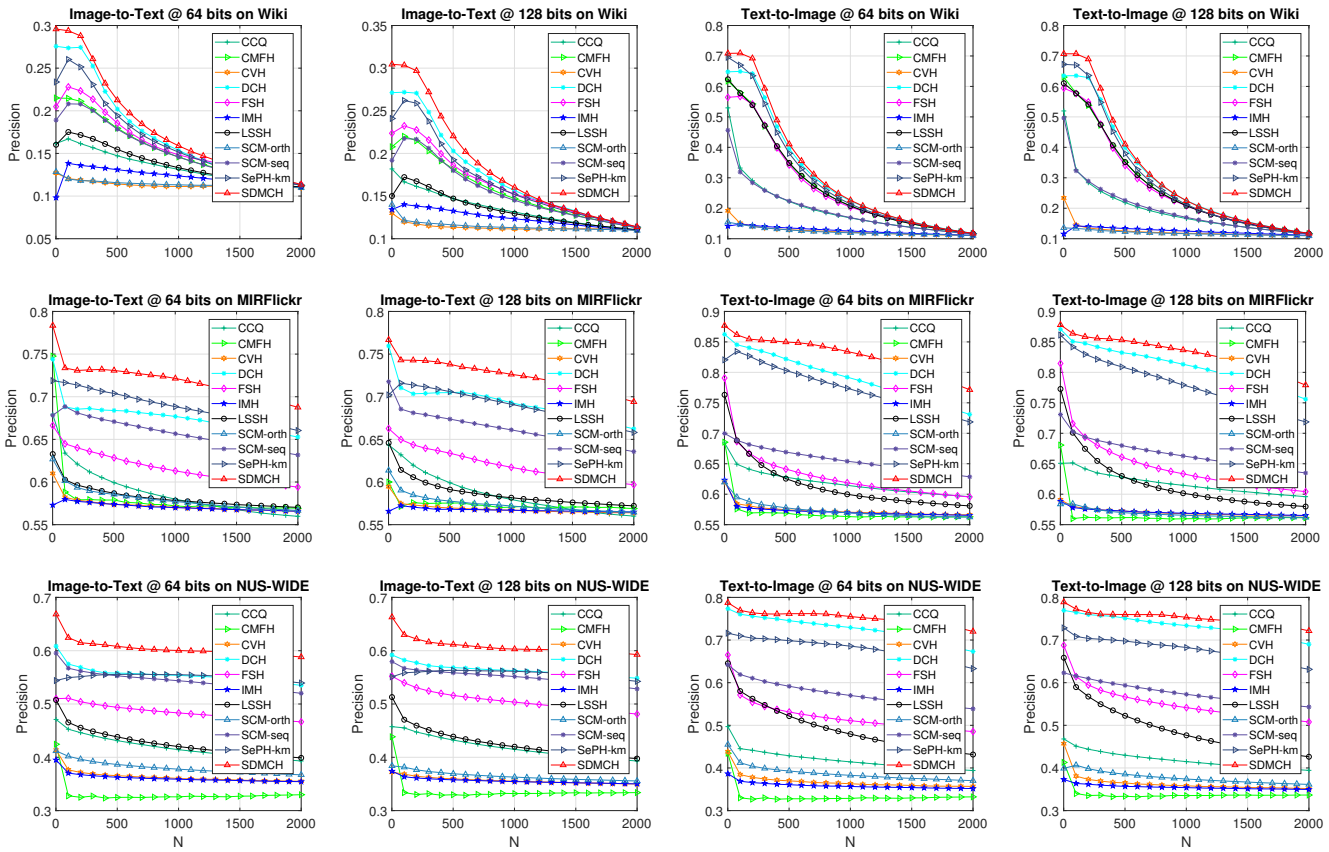

Figure 1: The top-N precision curves of various models on Wiki, MIRFlickr, and NUS-WIDE.

SDMCH, perform better than unsupervised ones, demonstrating the effectiveness of full utilization of the semantic information.

Top-N Precision Curves: Figure 1 illustrates the top-N precision curves in the cases of 64-bit and 128-bit code lengths on the three datasets. From this figure, we have the following observations. First, the curve of SDMCH is much higher than those of the other baselines in most cases, well demonstrating its superiority. Second, SDMCH, DCH, SePH-km, SCM-seq, and FSH are in the first group, performing better than the other baselines. Third, supervised methods, such as DCH, SePH-km, and SDMCH, generally perform better than unsupervised ones, and this phenomenon is consistent with the results in Table 1.

Precision-Recall Curves: The precision-recall curves in the cases of 64-bit and 128-bit code lengths are plotted in Figure 2. We have the similar observations to those with respect to the MAP and Top-N precision curves. For example, SDMCH always performs the best and performs much better than the baselines in some cases; generally, supervised ones also perform better than unsupervised ones.

In summary, SDMCH can achieve the state-of-the-art accuracy, demonstrating that embedding both the manifold structure and semantic information can ensure more precise hash codes. It also verifies the effectiveness of the loss function and optimization algorithm of SDMCH.

4.5 Convergence Analysis

We further made empirical analysis of its convergence by experiments on the three datasets with the hash code length being 128-bit. The curves of the objective values on the three datasets are plotted in Figure 3. It is worth noting that, for convenience, the objective values are normalized by dividing the maximums on each dataset. From this figure, we can observe that SDMCH can converge quickly. For example, it converges within 10 iterations on the three datasets, well demonstrating the effectiveness of the loss function and the iterative optimization algorithm.

4.6 Time Cost Analysis

The time complexity of solving Eq. (8), Eq. (10) and Eq. (12) is $O(nr+ncr+r^2+r^2c+r^3+nr^2)$, $O(nr^2+r^3+nrc+nr^2)$, and $O(nr+nr)$, respectively. Since $c, r \ll n$, the overall computational complexity of these steps is linear to the size of training data n. Since we can precompute $\mathbf{A}$ and $\mathbf{A}$ is a sparse matrix, the computation of Eq. (6) is also very efficient.

To verify the efficiency of SDMCH, we further compared the training time (in seconds) of all methods on MIRFlickr. The results are summarized in Table 2 and some observations can be found from these results. First, unlike some baslines, e.g., CCQ, SePH-km, and DCH, the training time of SDMCH does not significantly increase when the code length becomes longer. Second, some baselines, such as CMFH, CVH, and

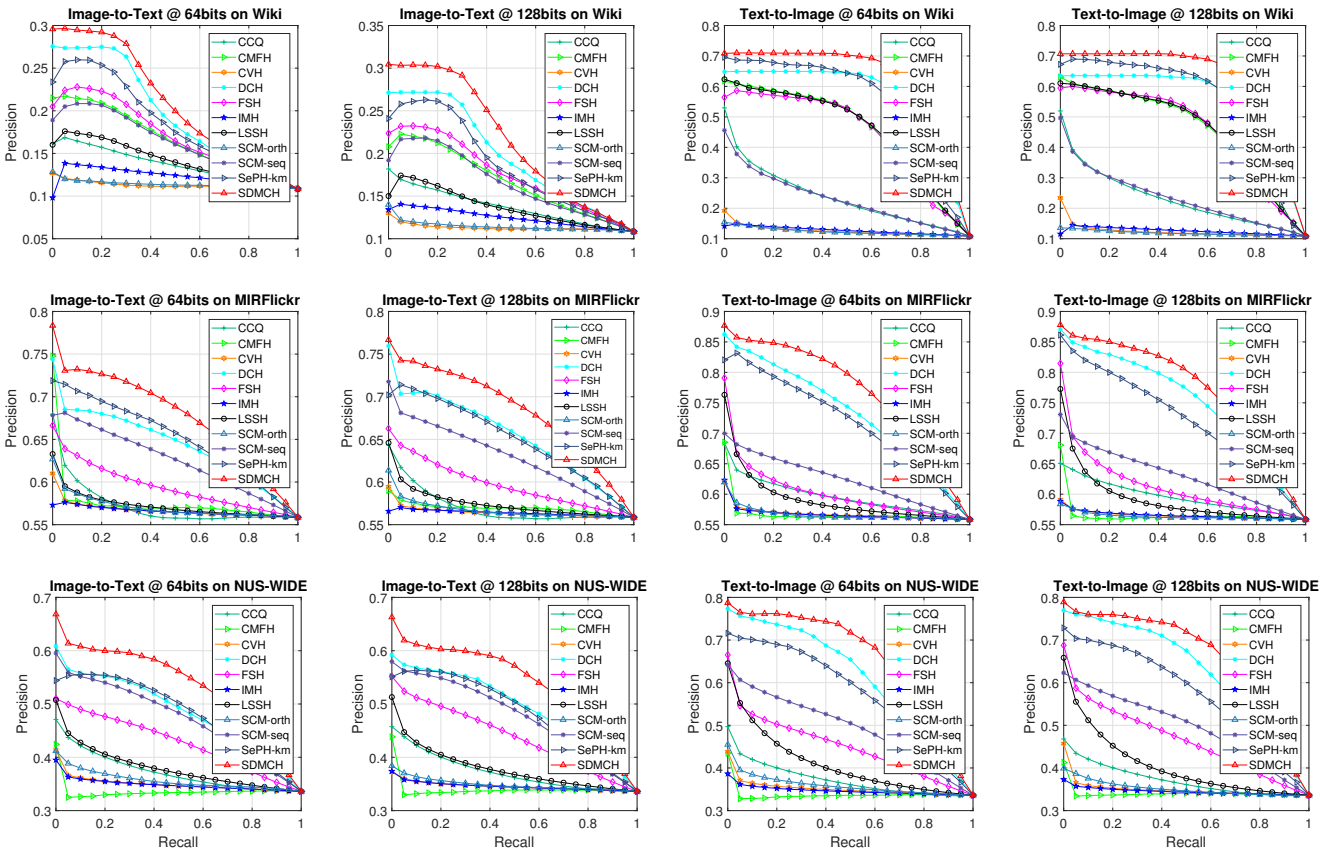

Figure 2: The precision-recall curves of various models on Wiki, MIRFlickr, and NUS-WIDE.

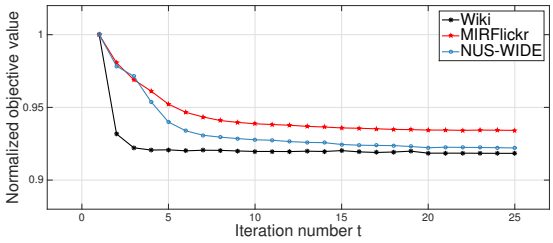

Figure 3: The convergence curves in the case of 128-bit code length on three datasets.

Method	32 bits	64 bits	96 bits	128 bits
CCQ	3.96	10.99	18.79	43.44
CMFH	0.12	0.14	0.17	0.19
CVH	0.06	0.05	0.05	0.05
DCH	10.21	32.37	69.81	115.16
FSH	9.59	10.94	12.61	12.43
IMH	13.32	13.22	13.56	13.44
LSSH	9.71	10.50	11.73	12.80
SCM-orth	0.04	0.04	0.04	0.04
SCM-seq	0.38	0.78	1.22	1.55
SePH-km	398.09	434.22	614.01	713.53
SDMCH	9.15	9.25	9.50	9.66

Table 2: The training time of various models on MIRFlickr.

SCM-seq, can be trained fast. However, as shown in the above Section 4.4, their retrieval accuracy is much worse than that of SDMCH. Third, DCH is competitive with SDMCH on the accuracy as shown in Table 1; however, it needs much more training time than that of SDMCH, especially in the cases of longer code lengths.

Therefore, considering both the accuracy and training time, we can conclude that SDMCH is more practical than the baselines.

5 Conclusion and Future Work

In this paper, we present a novel supervised cross-modal hashing method, named Supervised Discrete Manifold-Embedded Cross-Modal Hashing, SDMCH for short. With preserving the similarity constructed by manifold learning, SDMCH can exploit the non-linear manifold structure of data and construct the correlation among heterogeneous multiple modalities. Moreover, it can fully utilize the semantic information and generate the binary codes discretely. Extensive experiments on the three benchmark datasets demonstrate that SDMCH outperforms ten state-of-the-art cross-modal hashing methods.

In this work, we implement SDMCH as a shallow model because we want to focus on the design of the loss function and optimization scheme. Recently, deep hashing models have demonstrated more promising results. In our future work, we plan to extend it to a deep hashing model.

Acknowledgments

This work was partially supported by the Key Research and Development Program of Shandong Province (2016GGX101044), the National Natural Science Foundation of China (61573212, 61772310, 61702300, 61702302), and the Project of Thousand Youth Talents 2016.

References

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: A real-world web image database from national university of singapore. In *CIVR*, page 48, 2009.

[Ding *et al.*, 2014] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2075–2082, 2014.

[Huiskes and Lew, 2008] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *MIR*, pages 39–43, 2008.

[Irie *et al.*, 2014] Go Irie, Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Locally linear hashing for extracting non-linear manifolds. In *CVPR*, pages 2115–2122, 2014.

[Ji *et al.*, 2017] Rongrong Ji, Hong Liu, Liujuan Cao, Di Liu, Yongjian Wu, and Feiyue Huang. Toward optimal manifold hashing via discrete locally linear embedding. *TIP*, 26(11):5411–5420, 2017.

[Jiang and Li, 2017] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. In *CVPR*, pages 3270–3278, 2017.

[Kumar and Udupa, 2011] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365, 2011.

[Lin *et al.*, 2013] Guosheng Lin, Chunhua Shen, David Suter, and Anton van den Hengel. A general two-step approach to learning-based hashing. In *ICCV*, pages 2552–2559, 2013.

[Lin *et al.*, 2015] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015.

[Liu *et al.*, 2011] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *ICML*, pages 1–8, 2011.

[Liu *et al.*, 2015] Xianglong Liu, Lei Huang, Cheng Deng, Jiwen Lu, and Bo Lang. Multi-view complementary hash tables for nearest neighbor search. In *ICCV*, pages 1107–1115, 2015.

[Liu *et al.*, 2017] Hong Liu, Rongrong Ji, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Cross-modality binary code learning via fusion similarity hashing. In *CVPR*, pages 7380–7388, 2017.

[Long *et al.*, 2016] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S Yu. Composite correlation quantization for efficient multimodal retrieval. In *SIGIR*, pages 579–588, 2016.

[Luo *et al.*, 2018] Xin Luo, Liqiang Nie, Xiangnan He, Ye Wu, Chen Zhen-Duo, and Xin-Shun Xu. Fast scalable supervised hashing. In *SIGIR*, 2018.

[Masci *et al.*, 2014] Jonathan Masci, Michael M. Bronstein, Alexander M. Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *TPAMI*, 36(4):824–830, 2014.

[Rasiwasia *et al.*, 2010] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *MM*, pages 251–260, 2010.

[Roweis and Saul, 2000] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Shen *et al.*, 2015] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, Zhenmin Tang, and Heng Tao Shen. Hashing on nonlinear manifolds. *TIP*, 24(6):1839–1851, 2015.

[Song *et al.*, 2011] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *MM*, pages 423–432, 2011.

[Song *et al.*, 2013] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, pages 785–796, 2013.

[Tang *et al.*, 2016] Jun Tang, Ke Wang, and Ling Shao. Supervised matrix factorization hashing for cross-modal retrieval. *TIP*, 25(7):3157–3166, 2016.

[Weiss *et al.*, 2009] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2009.

[Wu *et al.*, 2014] Fei Wu, Zhou Yu, Yi Yang, Siliang Tang, Yin Zhang, and Yueting Zhuang. Sparse multi-modal hashing. *TMM*, 16(2):427–439, 2014.

[Xu *et al.*, 2017] Xing Xu, Fumin Shen, Yang Yang, Heng Tao Shen, and Xuelong Li. Learning discriminative binary codes for large-scale cross-modal retrieval. *TIP*, 26(5):2494–2507, 2017.

[Xu, 2016] Xin-Shun Xu. Dictionary learning based hashing for cross-modal retrieval. In *MM*, pages 177–181, 2016.

[Yang *et al.*, 2008] Yi Yang, Yueting Zhuang, Fei Wu, and Yunhe Pan. Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval. *TMM*, 10(3):437–446, 2008.

[Yang *et al.*, 2017a] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*, pages 1618–1625, 2017.

[Yang *et al.*, 2017b] Rui Yang, Yuliang Shi, and Xin-Shun Xu. Discrete multi-view hashing for effective image retrieval. In *ICMR*, pages 175–183, 2017.

[Zhang and Li, 2014] Dongqing Zhang and Wu-Jun Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, pages 2177–2183, 2014.

[Zhang *et al.*, 2011] Dan Zhang, Fei Wang, and Luo Si. Composite hashing with multiple information sources. In *SIGIR*, pages 225–234, 2011.

[Zhang *et al.*, 2015] Lei Zhang, Yongdong Zhang, Richang Hong, and Qi Tian. Full-space local topology extraction for cross-modal retrieval. *TIP*, 24(7):2212–2224, 2015.

[Zhang *et al.*, 2017] Peng-Fei Zhang, Chuan-Xiang Li, Meng-Yuan Liu, Liqiang Nie, and Xin-Shun Xu. Semi-relaxation supervised hashing for cross-modal retrieval. In *MM*, pages 1762–1770, 2017.

[Zheng *et al.*, 2017] Feng Zheng, Yi Zheng, and Ling Shao. Hetero-manifold regularisation for cross-modal hashing. *TPAMI*, PP(99):1–1, 2017.

[Zhou *et al.*, 2014] Jile Zhou, Guiguang Ding, and Yuchen Guo. Latent semantic sparse hashing for cross-modal similarity search. In *SIGIR*, pages 415–424, 2014.