Deep Unified Cross-Modality Hashing by Pairwise Data Alignment

Yimu Wang, Bo Xue, Quan Cheng, Yuhui Chen and Lijun Zhang *
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{wangym, xueb, chengq, chenyuhui, zhanglj}@lamda.nju.edu.cn

Abstract
With the increasing amount of multimedia data, cross-modality hashing has made great progress as it achieves sub-linear search time and low memory space. However, due to the huge discrepancy between different modalities, most existing cross-modality hashing methods cannot learn unified hash codes and functions for modalities at the same time. The gap between separated hash codes and functions further leads to bad search performance. In this paper, to address the issues above, we propose a novel end-to-end Deep Unified Cross-Modality Hashing method named DUCMH, which is able to jointly learn unified hash codes and unified hash functions by alternate learning and data alignment. Specifically, to reduce the discrepancy between image and text modalities, DUCMH utilizes data alignment to learn an auxiliary image to text mapping under the supervision of image-text pairs. For text data, hash codes can be obtained by unified hash functions, while for image data, DUCMH first maps images to texts by the auxiliary mapping, and then uses the mapped texts to obtain hash codes. DUCMH utilizes alternate learning to update unified hash codes and functions. Extensive experiments on three representative image-text datasets demonstrate the superiority of our DUCMH over several state-of-the-art cross-modality hashing methods.

1 Introduction
With the increasing multimedia data, approximate nearest neighbor (ANN) search has been a fundamental problem in the information retrieval area. Among several ANN search methods, hashing [Wang et al., 2018; Chen et al., 2019; Wang et al., 2020] has attracted extensive attention. It maps data points to binary codes with hash functions by preserving the similarity in the original space of data points. Due to binary codes, the storage cost can be dramatically reduced with sub-linear search complexity. In many applications, such as search engines, systems have to handle data of multi-modalities. It requires researchers to support cross-modality retrieval that can return relevant results of one modality when querying another modality, e.g., retrieving images with text queries. Due to its low storage cost and search time, cross-modality hashing (CMH) area receives more and more attention recently.

Existing CMH methods can be roughly divided into shallow CMH methods [Zhai et al., 2013; Wu et al., 2015; Xie et al., 2016; Ye and Peng, 2018] and deep CMH methods [Jiang and Li, 2017; Yan et al., 2017; Li et al., 2018; Shi et al., 2019; Sun et al., 2019; Xu et al., 2019]. Most shallow methods capture the semantic relevance in a common Hamming space and learn hash functions that map hand-crafted features to hash codes, which means the feature extraction procedure is independent of the hash code learning procedure. That may block the way to achieve satisfying performance, as the hand-crafted features might not be optimal for the hash code learning procedure. Recently, deep learning [Zagoruyko and Komodakis, 2016] has shown its superiority of representation learning in various applications, such as image recognition [He et al., 2016]. Deep CMH methods leverage the power of deep learning by integrating feature learning and hash code learning into a single framework. As a result, deep methods capture non-linear correlations among cross-modality instances and achieve better performance than shallow methods. The key to CMH is to bridge the modality gap, as it makes original data distributions and feature representations of modalities different. Thus, it is necessary to explore semantic relevance between different modalities in sufficient details to reduce that gap and further improve search performance. Recently, some supervised deep methods [Li et al., 2018; Zhan et al., 2020] exploit semantic labels or relevance information, thereby better cross-modality correlations.

However, previous work [Jiang and Li, 2017; Li et al., 2018; Sun et al., 2019] mainly aligns different modalities in the level of representation using pre-defined loss functions and derives independent hash functions and codes for different modalities. Representation (level) alignment may reduce the modality gap in some intermediate procedures of independent hash functions, but inputs, other layers and final outputs (hash codes) of deep hash functions are usually of different distributions. Actually, high-dimensional modality-specific original data contain abundant information that enables us to bridge the modality gap by the data (level) alignment. As we have paired data of different modalities, e.g., image-text paired data, leveraging that data to learn mappings between
modalities (data alignment) is possible and may bridge the modality gap better than representation alignment. Besides, with the mappings between modalities, we can derive unified hash functions and learn unified hash codes for correlated data for different modalities at the same time, while existing work cannot. The difference between data alignment (unified hash codes and functions) and representation alignment (separated hash codes and functions) is presented in Figure 1.

In this paper, to better bridge the modality gap and derive unified hash functions and hash codes for different modalities, we propose a novel Deep Unified Cross-Modality Hashing (DUCMH) method. Specifically, DUCMH alternatively learns unified hash functions and unified hash codes, while aligning different modalities by data alignment under the supervision of image-text pairs at hand. Our unified hash function maps texts to hash codes, while with an auxiliary image to text mapping trained under the supervision of paired data, the unified hash function is also able to map images to hash codes. The auxiliary image to text mapping maximizes the semantic relevance and distribution consistency between different modalities, while the unified hash function is employed for unifying hash codes of different modalities and further reducing modality gap. To empirically evaluate our DUCMH, we conduct extensive experiments on three commonly used image-text datasets, showing the superiority of our method over several state-of-the-art methods.

The main contributions of DUCMH are summarized below:

- **To the best of our knowledge, DUCMH is the first deep method that derives unified hash codes for database instances and unified hashing functions for unseen query points at the same time.**
- **DUCMH is the first deep method to apply data alignment and learn the mapping between modalities. Data alignment can carefully preserve semantic relevance and distribution consistency across modalities, while effectively bridging the modality gap.**
- **Extensive experiments conducted on three image-text benchmarks show the superiority of our DUCMH over several state-of-the-art cross-modality hashing methods, including both shallow and deep methods.**

## 2 Related Work

Existing cross-modality hashing work can be divided into unsupervised and supervised methods. For unsupervised methods, the intra- and inter-modality relations are exploited to generate hash codes without any supervised information [Hotelling, 1992]. On the other side, with more semantic information, supervised methods are able to capture correlations better and thus achieve superior performance [Ding et al., 2014; Zhang and Li, 2014; Wang et al., 2015; Luo et al., 2018]. Semantic Correlation Maximization (SCM) [Zhang and Li, 2014] reduces the modality gap by utilizing label information to learn a modality-specific transformation, and preserves the maximal correlation between modalities. Collective Matrix Factorization Hashing (CMFH) [Ding et al., 2014] employs the latent factor model to learn hash codes by collective matrix factorization from different modalities. Semantic Topic Multimodal Hashing (STMH) [Wang et al., 2015] utilizes latent semantic information among different modalities to learn hash codes. Supervised Discrete Manifold-Embedded Cross-Modality Hashing (SDMCH) [Luo et al., 2018] generates binary hash codes by exploiting the non-linear manifold structure of data and constructs the correlations among heterogeneous multiple modalities with semantic information.

The above methods utilize hand-crafted features to generate hash codes, which might not be optimally compatible with the hash-code learning procedure. With the advances of deep learning, more and more deep supervised CMH work has been proposed. Deep methods [Jiang and Li, 2017; Li et al., 2018; Shi et al., 2019; Zhan et al., 2020] integrate feature learning and hash-code learning into a unified architecture with promising performance. Deep Cross-modality Hashing (DCMH) [Jiang and Li, 2017] is the first deep framework that performs feature learning and hash-code learning simultaneously. Self-Supervised Adversarial Hashing (SSAH) [Li et al., 2018] incorporates a self-supervised semantic network coupled with multi-label information, and carries out adversarial learning to maximize the semantic relevance and feature distribution consistency between different modalities. Equally-Guided Discriminative Hashing (EGDH) [Shi et al., 2019] takes semantic structure and discriminability into consideration to learn better hash functions and hash codes. Supervised
Hierarchical Deep Hashing (SHDH) [Zhan et al., 2020] explores the power of hierarchical labels to further boost the learning procedure.

While previous methods achieve success in performance by representation alignment, they ignore image-text pairs at hand, and never leverage that information to reduce the modality gap by data alignment or derive unified hash functions and codes simultaneously. To the best of our knowledge, DUCMH is the first deep method to employ data alignment by fully utilizing the image-text paired data for better reducing the modality gap and deriving unified hash functions and codes simultaneously.

3 DUCMH

In this section, we present the details about our deep unified cross-modality hashing (DUCMH) method, including model formulation and learning algorithm. DUCMH, shown in Figure 2, is an end-to-end learning framework consisting two parts: the feature learning part and the hash-code learning part. We employ asymmetric learning to alternatively update each part and data alignment to bridge the modality gap.

3.1 Problem and Notation

Scalars are defined by lowercase letters, such as \( w \). Vectors and matrices are denoted by boldface lowercase letters and uppercase letters, e.g., \( \mathbf{w} \) and \( \mathbf{W} \). The \( i \)-th element of \( \mathbf{w} \) and \( \mathbf{W} \) is represented by subscripts as \( w_i \) and \( W_{ij} \). Besides, \( W_{ij} \) and \( W_{ix} \) represent the \( j \)-th column and \( i \)-th row of matrix \( \mathbf{W} \), while \( W_{ix} \) and \( W_{ij} \) represent the matrices of matrix \( \mathbf{W} \) without \( j \)-th column and \( i \)-th row.

Without loss of generality, we focus on CMH using image-text paired data. Assume that we have \( n \) training data points, in which each instance has two modalities of features, i.e., image and text. \( O = \{\mathbf{o}_i\}_{i=1}^n \) is a cross-modality dataset with \( n \) instance, while for \( i \)-th instance \( \mathbf{o}_i = (x_i, y_i, l_i) \), \( x_i \), \( y_i \), and \( l_i = [l_{i,1}, \ldots, l_{i,clsnum}] \) are the corresponding image, text and label, and \( clsnum \) is the number of classes. If \( o_i \) belongs to the \( j \)-th class, \( l_{i,j} = 1 \), otherwise \( l_{i,j} = 0 \). Besides, we denote \( L = [l_1, \ldots, l_n] \in \{0,1\}^{n \times clsnum} \). The similarity matrix \( S \in \{-1,+1\}^{n \times n} \) is generated by labels. We let \( S_{ij} = 1 \) if \( o_i \) and \( o_j \) are semantically similar, in another word, share at least one label, otherwise \( S_{ij} = -1 \).

Given the above training information \( O \) and \( S \), the goal of cross-modality hashing is to learn two hash functions \( h_x(\cdot) \in \{-1,+1\}^c \) for the image modality and \( h_y(\cdot) \in \{-1,+1\}^c \) for the text modality, where \( c \) is code length. Meanwhile, the similarity-preserving hash codes \( B = [b_1, \ldots, b_n]^\top \in \{-1,+1\}^{n \times c} \) should also be derived. Two hash functions are used for generating hash codes of the unseen data points while hash codes should preserve the cross-modality similarity in \( S \). Specifically, when \( S_{ij} = 1 \), the Hamming distance between the binary codes \( b_i \) and \( b_j \) should be small. Otherwise, the distance should be large.

3.2 Our Model – Learning by Paired Data

Feature Learning – Unified Hash Functions

As aforementioned, previous work [Li et al., 2018; Shi et al., 2019] usually employs representation alignment, which only reduces the modality gap in some intermediate procedure of algorithms. On the other side, as the image-text pairs of database are at hand, it is possible to employ data alignment to directly bridge the modality gap by learning a mapping from image to text or from text to image.

DUCMH chooses to learn the mapping \( f_{2t1}(\cdot) \) from image to text. There are two reasons: first, it is easier to learn the mapping from image to text than another, as the image always contains more information, such as spatial information and styles, which are hard to be pictured by a sentence. Second, images might be influenced a lot by the lights, angles, and other conditions, while texts are more steady and the light, angle, spatial information of an image can be easily represented by a simple word. That mapping \( f_{2t1}(\cdot) \) can be learned under the supervision of image-text pairs.

With the mapping \( f_{2t1}(\cdot) \), we can represent an image \( x \) as a text \( f_{2t1}(x) \). By employing an alignment loss presented in the following section, the modality gap between image and text can be better reduced than the representation alignment previous work employs.
For the real text $y$ and predicted text $f_{i2t}(x)$, we use a unified hash function $h_y(\cdot)$ mapping them to hash codes as,

\[ b^x = \text{sign}(h_x(x)) = \text{sign}(h_y(f_{i2t}(x))) , \]
\[ b^y = \text{sign}(h_y(y)) . \]

To generate discrete binary hash codes, the sign function $\text{sign}(\cdot)$ is attached after the last layer of our unified hash function $h_y(\cdot)$. However, as the gradients of $\text{sign}(\cdot)$ are $0$ at most of points, we use $\tanh(\cdot)$ instead to better train the unified hash function $h_y(\cdot)$ and the image to text mapping $f_{i2t}(\cdot)$.

**Hash Code Learning – Unified Hash Codes**

In this section, we present our objective function. As unified hash codes $B$ should preserve semantic information and also be capable of classification, we design three corresponding losses shown below. Besides, to bridge the modality gap, we also employ an alignment loss to align different modalities for better search performance.

First, DUCMH leverages image-text pairs $\{(x_i,y_i)\}_{i=1}^n$ to learn a precise image to text mapping $f_{i2t}(\cdot)$ for data alignment. Thus, to fully bridge the modality gap on the database points, we use alignment loss to model the modality gap, which is shown below,

\[ \ell_{\text{align}} = \epsilon \sum_{i=1}^n \| f_{i2t}(x_i) - y_i \|_2 . \]  

(1)

where $\epsilon$ is a hyper-parameter.

Second, as the learned database hash codes and the hash codes generated by the hashing function should preserve semantic similarity, we employ the similarity loss [Zhang and Li, 2017] is as:

\[ \ell_{\text{sim}} = \sum_{i=1}^n \sum_{j=1}^n \| h_x(x_i)b_j^x - cS_{ij} \|^2 \]
\[ + \sum_{i=1}^n \sum_{j=1}^n \| h_y(y_i)b_j^y - cS_{ij} \|^2 \]
\[ + \alpha \sum_{i=1}^n \sum_{j=1}^n \| h_x(x_i)h_y(y_i)^\top - cS_{ij} \|^2 , \]

(2)

where $\alpha$ is a hyper-parameter, $b_i$ and $b_j$ are the hash codes of $i$-th and $j$-th data point learned by the alternative learning algorithm shown in the following section.

Third, as the hash space is small, i.e., $2^c$ points, we expect hash codes are able to preserve discriminative semantic information and recover labels with simple transformations. We use $h_x(\cdot)W$ and $bW$ to represent the predicted label recovered from hash codes, where $W \in \mathbb{R}^{c \times const}$. To simplify the formulation of this loss, we employ Mean Squared Error to measure the label information contained in hash codes. Now, the classification loss is defined as follows:

\[ \ell_{\text{cls}} = \alpha \sum_{i=1}^n \left( \| h_x(x_i)W - l_i \|^2 + \| h_y(y_i)W - l_i \|^2 \right) \]
\[ + \alpha \sum_{i=1}^n \| b_iW - l_i \|^2 + \| W \|^2_F , \]

(3)

where $\alpha$ is a hyper-parameter defined above.

Last but not the least, as we replace $\text{sign}(\cdot)$ with $\tanh(\cdot)$ after the last layer of our unified hash function $h_y(\cdot)$ for better performance, hash functions is still expected to generate discrete hash codes. Intuitively, the quantization loss [Jiang and Li, 2017] is as:

\[ \ell_{\text{quan}} = \rho \sum_{i=1}^n (\| h_x(x_i) - b_i \|^2 + \| h_y(y_i) - b_i \|^2) , \]

(4)

where $\rho$ is a hyper-parameter.

Now, by minimizing four losses, our objective function can be formulated as:

\[ \min_{\Theta, B, W} \ell = \ell_{\text{align}} + \ell_{\text{sim}} + \ell_{\text{cls}} + \ell_{\text{quan}} , \]

(5)

where $\Theta$ is the parameter of the unified hash function $h_y(\cdot)$ and the image to text mapping $f_{i2t}(\cdot)$. Hyper-parameters $\epsilon$, $\alpha$ and $\rho$ are empirically set to 5000, 50 and 200 for scaling the order of each loss.

### 3.3 Optimization

In this part, we present an alternative learning algorithm to learn $\Theta$, $B$ and $W$ in Eq. (5), which means we update one parameter with others fixed. This algorithm is also summarized in Algorithm 1, while details are presented below.

Normally, we are only given database points $O$ and the pairwise supervised information $S$ between them. To accelerate the training procedure, we can learn hash-codes and hash functions by sampling a subset of $O$ as the query set $O_{\text{train}}$ for training, i.e., $O_{\text{train}} \subseteq O$ in each step. Thus, the training complexity can be reduced to $O(\text{nm})$ from $O(n^2)$, where $O$ hides variables irrelevant to the size of dataset, $m$ and $n$ represent the size of datasets $O$ and $O_{\text{train}}$.

**Learn $\Theta$ with $B$ and $W$ fixed**

When $B$ and $W$ are fixed, we learn and update the parameter $\Theta$ by back-propagation algorithm.

**Learn $B$ with $\Theta$ and $W$ fixed**

To simplify the calculation, when fixing $\Theta$ and $W$, we rewrite Eq. (5) in the matrix form as:

\[ \min_B \ell = \| VBT - cS \|^2_F + \| TB^T - cS \|^2_F \]
\[ + \rho (\| V - B \|^2_F + \| T - B \|^2_F ) \]
\[ + \alpha \| BW - L \|^2_F + \text{const} , \]

where $\text{const}$ is the constant independent of $B$, $V = [h_x(x_1)^\top, \ldots, h_x(x_n)^\top] \in \mathbb{R}^{n \times c}$, $T = [h_y(y_1)^\top, \ldots, h_y(y_n)^\top] \in \mathbb{R}^{n \times c}$.

With little algebra, we can derive,

\[ \min_B \ell = \| VBT \|^2_F + \| TB^T \|^2_F + \alpha \| BW \|^2_F \]
\[ - 2\alpha \text{tr} (BV^T S) - 2\alpha \text{tr} (BT^T S) \]
\[ - 2 \alpha \text{tr} (BWLT) - 2 \rho \text{tr} (B(T + V)^\top) + \text{const} \]
\[ = \text{tr}(B(Q + R)) + \text{const} , \]

where $Q = V^TBV^T + T^TB^T + \alpha WW^TB^T$ and $R = -2cV^T S - 2cT^T S - 2cWL^T - 2\rho(T + V)^\top$, and $\text{tr}(\cdot)$ is the trace of a matrix.
As this problem is NP-hard, to ease this problem, inspired by SDH [Shen et al., 2015], we update one bit a time using the discrete cyclic coordinate descent method. We alternatively update a column of $B$ with other columns fixed. Hence, this bit by bit problem at bit $k$ becomes:

$$
\min_{B_{sk}} \ell = \text{tr}(B_{sk}(Q_{ks} + R_{ks})) + \text{const},
$$

(6)

where $Q_{ks} = V_s^T B_{sk} V_k^T + T_s^T B_{sk} T_k^T + \alpha W_k W_k^T B_{sk}^T$, $R_{ks} = -2e_k V_k^T + 2T_k^T S - 2\alpha W_k L - 2\rho(T_s + V_k)^T$. Next, the optimal solution of problem (6) is as follows:

$$
B_{sk} = -\text{sign}(Q_{ks} + R_{ks})^T.
$$

(7)

Learn $W$ with $\Theta$ and $B$ fixed

When fixing $\Theta$ and $B$, we rewrite Eq. (5) as:

$$
\min_W \ell = \alpha \left( ||V W - L||_F^2 + ||T W - L||_F^2 \right) + \alpha ||B W - L||_F^2 + ||W||_F^2 + \text{const},
$$

where const is the constant independent of $W$. It is easy to solve $W$ by the regularized least squares problem, while the closed-form solution is as:

$$
W = P^{-1}(\alpha V + \alpha T + \alpha B^T B + I),
$$

(8)

where $P = (\alpha V^T V + \alpha T^T T + \alpha B^T B + I)$ and $I$ is the diagonal with elements being 1.

4 Experiment

In this section, we carry out experiments to empirically evaluate the performance of DUCMH on three image-text benchmarks, and then compare it to state-of-the-art approaches.
4.3 Results on Image and Text (Tags)

Following the experiment settings of previous work [Ding et al., 2014; Jiang and Li, 2017; Li et al., 2018; Shi et al., 2019], we first test the power of data alignment between text (tags) and image modalities.

The mAP searching results are presented in Table 1. In all datasets and two tasks, i.e., text to image (T → I) and image to text (I → T), our proposed DUCMH outperforms all the methods with the power of data alignment and unified hash functions and codes. Specifically, in MIRFLICKR-25K, DUCMH outperforms SOTAs by 0.0461, 0.0423, 0.0325 of 16, 32, and 64 bits on the task of T → I, respectively, while on the task of I → T, our DUCMH outperforms SOTAs by 0.0895, 0.0844, 0.0770 of 16, 32, and 64 bits. Similar results can be found in other datasets on two tasks.

4.4 Results on Image and Text (Sentences)

As the original data of text modality in IAPR TC-12 are sentences, we conduct experiments on IAPR TC-12 to test the power of data alignment between text (sentences) and image modalities. We replace the image to text mapping $f_{2t}()$ with a classical image caption method, i.e., show and tell [Vinyals et al., 2015], and the unified hash function $h_y(·)$ with a classical sentence classification method, i.e., TextCNN [Kim, 2014] (represented by “DUCMH with stcs” in Table 2). We still use the hyper-parameters introduced in the original papers, while other hyper-parameters $\epsilon, \alpha$ and $\beta$ are still the same.

The results are shown in Table 2. As tags are pure semantic information without noises extracted from sentences and learning to predict conceptual tags is much easier than learning to caption images, it is reasonable that the model using tags (“DUCMH with tags” in Table 2) outperforms the model using sentences (“DUCMH with stcs” in Table 2).

<table>
<thead>
<tr>
<th>Task</th>
<th>Method</th>
<th>MIRFLICKR-25K</th>
<th>IAPR TC-12</th>
<th>NUS-WIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16 bits</td>
<td>32 bits</td>
<td>64 bits</td>
</tr>
<tr>
<td>T → I</td>
<td>DUCMH with tags</td>
<td>0.6007</td>
<td>0.6579</td>
<td>0.6880</td>
</tr>
<tr>
<td></td>
<td>DUCMH with stcs</td>
<td>0.5824</td>
<td>0.5909</td>
<td>0.6140</td>
</tr>
<tr>
<td>I → T</td>
<td>DUCMH with tags</td>
<td>0.6392</td>
<td>0.6908</td>
<td>0.7104</td>
</tr>
<tr>
<td></td>
<td>DUCMH with stcs</td>
<td>0.6106</td>
<td>0.6479</td>
<td>0.6659</td>
</tr>
</tbody>
</table>

Table 2: Comparison of mAP w.r.t. different number of bits on IAPR TC-12. “stcs” represents sentences.

Acknowledgments

This work was partially supported by the NSFC (61976112), JiangsuSF (BK20200064), and Open Research Projects of Zhejiang Lab (NO. 2021KB0AB02).
References


