2021 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-3864-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICME51207.2021.9428323

EFFICIENT ONLINE LABEL CONSISTENT HASHING FOR LARGE-SCALE CROSS-MODAL RETRIEVAL

Jinhan Yi[1,2,5], *Xin Liu*[1,2,*], *Yiu-ming Cheung*[3], *Xing Xu*[4], *Wentao Fan*[1,2], *Yi He*[1,2]

[1] Department of Computer Science and Technology, Huaqiao University, Xiamen, 361021, China
[2] Xiamen Key Lab. of Computer Vision and Pattern Recognition, Fujian Key Lab. of Big Data Intelligence and Security, China
[3] Department of Computer Science, Hong Kong Baptist University, Kowloon, Hong Kong
[4] School of Computer Science and Engineering, University of Electronic Science and Technology of China
[5] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University
Email: {jhyi, xliu, fwt}@hqu.edu.cn, ymc@comp.hkbu.edu.hk, xing.xu@uestc.edu.cn. * Corresponding author.

ABSTRACT

Existing cross-modal hashing still faces three challenges: (1) Most batch-based methods are unsuitable for processing large-scale and streaming data. (2) Current online methods often suffer from insufficient semantic association, while lacking flexibility to learn the hash functions for varying streaming data. (3) Existing supervised methods always require much computation time or accumulate large quantization loss to learn hash codes. To address above challenges, we present an efficient Online Label Consistent Hashing (OLCH) for cross-modal retrieval, which aims to incrementally learn hash codes for the current arriving data, while updating the hash functions at a streaming manner. To be specific, an online semantic representation learning framework is designed to adaptively preserve the semantic similarity across different modalities, and a mini-batch online gradient descent approach associated with forward-backward splitting is developed to optimize the hash functions. Accordingly, the hash codes are adaptively learned online with the high discriminative capability, while avoiding high computation complexity to process the streaming data. Experimental results show its outstanding performance in comparison with the-state-of-arts.

Index Terms— Online label consistent hashing, online semantic representation, mini-batch online gradient descent, forward-backward splitting

1. INTRODUCTION

With the explosive growth of various kinds of multimedia data on the Internet, cross-modal retrieval has recently attracted wide attention. In particular, hashing has been widely applied in cross-modal retrieval due to its advantages of low storage and fast query [1, 2], and existing cross-modal hashing (CMH) methods can be divided into unsupervised learning [3, 4, 5, 6] and supervised learning [7, 8] cases, respectively, by ignoring and embedding the label information. Remarkably, the label information is able to well correlate the semantic information between different modalities, and often produce more compact hash codes. Nevertheless, these methods mainly learn the hash codes in batch-based manner, and require all training data points to be available when leaning the hash functions. On the one hand, multimedia data points in real application often continuously arrive in a streaming fashion. Under such circumstance, if the training data is increasingly accumulated, these batch-based methods have to recalculate the hash functions on the whole database, which is computationally inefficient. On the other hand, if the accumulated data is very large, it is impractical to load all the data into memory for hash function learning.

In recent years, some online CMH methods have been developed to cope with the streaming data [9, 10]. These methods select to update the hash functions from sequentially arriving data, and simultaneously encode these new data into compact binary codes. Although impressive performance has been achieved within these online learning strategies, these methods often suffer from the insufficient semantic association between heterogeneous modalities and may therefore weaken the semantic representation in the learnt hash codes.

In this paper, we propose an efficient Online Label Consistent Hashing (OLCH) approach to benefit cross-modal retrieval for streaming multi-modal data, and the main contributions are three-fold: 1) A multi-class classification scheme is utilize to supervise the semantic representation learning, which can well retain the label consistency and preserve the relative semantic similarity between heterogeneous data points. 2) An efficient online learning framework is developed for cross-modal retrieval, which can process the streaming

The work is supported by National Science Foundation of China (Nos. 61673185, 61876068 and 61976049), the National Science Foundation of Fujian Province (Nos. 2020J01083 and 2020J01084), the State Key Laboratory of Integrated Services Networks of Xidian University (No. ISN20-11), Quanzhou City Science & Technology Program of China (No. 2018C107R), IG-FNRA of HKBU with Grant: RC-FNRA-IG/18-19/SCI/03, and ITF of ITC of Hong Kong SAR under Project ITS/339/18.

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

data with low computational and memory costs. 3) A mini-batch online gradient descent approach is addressed to learn hash function adaptively, while a forward-backward splitting scheme is exploited to keep the sparsity of learnt hash code, leading to an efficient optimization and updating process. Extensive experiments show its outstanding performance.

2. RELATED WORK

In this section, we briefly survey the related works of CMH, including offline learning and online learning counterparts.

2.1. Offline CMH Fashion

Offline CMH can be broadly divided into unsupervised and supervised learning cases. Unsupervised CMH learns the unified hash codes directly from the paired training data so as to preserve the inter-modality and intra-modality information [3, 4, 5]. For instance, Collective Matrix Factorization Hashing (CMFH) [4] utilizes matrix factorization technique to learn the cross-view hash codes. Supervised CMH methods leverage the label information to promote the hash code learning [7, 11, 12, 13, 14, 8]. Along this line, Semantic Correlation Maximization (SCM) [11] utilizes the label information to maximize the semantic correlation. Supervised Matrix Factorization Hashing (SMFH) [15] adds the label supervision to perform collective matrix factorization. Generalized Semantic Preserving Hashing (GSePH) [13] constructs an affinity matrix in a supervised manner. Discrete Cross-modal Hashing (DCH) [8] directly updates hash codes bit by bit while retaining the discrete constraints. In recent years, deep-networks-based cross-modal hashing methods [7, 16] can handle the insufficient representation of the hand-crafted features more effectively. Although these methods are effective for cross-modal retrieval, they are batch based learning methods and therefore are unsuitable for the online scenario.

2.2. Online CMH Fashion

In the past few years, some online hash methods have been proposed [17, 18, 19]. It is noted that these methods are designed to process the data on single modality, which cannot be directly extended to cross-modal retrieval scenarios. To adapt multi-modal media data, Online Cross-Modal Hashing (OCMH) [9] is proposed to support online learning, which decomposes the hash codes matrix to a shared latent codes matrix and a transfer matrix. Online Collective Matrix Factorization Hashing (OCMFH) [20] utilizes the collective matrix factorization to learn the hash code in an online manner Nevertheless, these two methods select unsupervised learning mechanisms, and the derived hash codes are not discriminative enough for high retrieval accuracy. To embedding the label supervision, Online Latent Semantic Hashing (OLSH) [10] preserves the semantic correlations in the continuous latent semantic space, which often yields promising retrieval performances to process the streaming data points. Nevertheless, such method cannot discriminatively preserve the category information of tags in the learnt hash codes, and its retrieval performance is not very stable.

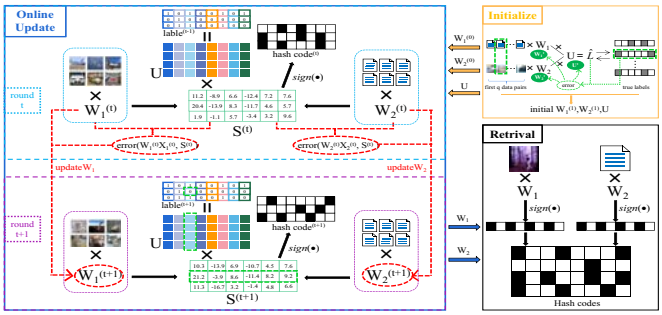


Fig. 1: Graphic illustration of OLCH Framework.

3. ONLINE LABEL CONSISTENT HASHING

This paper mainly focuses on bimodal data (i.e., image and text), which can be easily extended to more modalities.

3.1. Problem Description

Suppose the training set consists of multiple streaming image-text pairs. At each round t, a new data chunk $[\mathbf{X}_1^{(t)}, \mathbf{X}_2^{(t)}]$ of size N_t is added to the training set, where $\mathbf{X}_1^{(t)} \in \mathbb{R}^{d_1 \times N_t}$, $\mathbf{X}_2^{(t)} \in \mathbb{R}^{d_2 \times N_t}$ denote the feature matrices of image and text , d_1 and d_2 denote their dimensionalities, respectively. Let $\mathbf{Y}^{(t)} \in \{0,1\}^{c \times N_t}$ represents the label matrix, where c is the number of categories. Let N be the size number of all data that has been obtained, the total training data, consisting of old data and new data, is denoted as $\mathbf{X}_m = [\tilde{\mathbf{X}}_m^{(t-1)}, \mathbf{X}_m^{(t)}]$, $\mathbf{Y} = [\tilde{\mathbf{Y}}^{(t-1)}, \mathbf{Y}^{(t)}]$, where $\tilde{\mathbf{X}}_m^{(t-1)} \in \mathbb{R}^{d_m \times (N-N_t)}$, $\tilde{\mathbf{Y}}^{(t-1)} \in \mathbb{R}^{c \times (N-N_t)}$. Similarly, the hash code of total data is $\mathbf{B} = [\tilde{\mathbf{B}}^{(t-1)}, \mathbf{B}^{(t)}]$, where $\tilde{\mathbf{B}}^{(t-1)} \in \{-1,1\}^{r \times (N-N_t)}$, $\mathbf{B}^{(t)} \in \{-1,1\}^{r \times N_t}$, r is the hash code length.

3.2. Formulation

Fig. 1 depicts the learning framework of OLCH. The core problem is to learn the potential semantic associations between heterogeneous features. Due to the discrete constraint of labels, it is hard to model the semantic correlations among heterogenous data points [21]. In order to make full use of each category within the label, we utilize the multi-class classification to classify the common semantic vector to the semantic labels, and its expression can be formulated as:

$$\begin{aligned} \min_{\mathbf{S}^{(t)}} J_L(\mathbf{S}^{(t)}) &= ||\mathbf{U}\mathbf{S}^{(t)} - \mathbf{Y}^{(t)}||_L \\ s.t. \mathbf{Y}^{(t)} &\in \{0,1\}^{c \times N_t} \end{aligned} \tag{1}$$

where $||\cdot||_L$ is the *Log-likelihood Loss*, $\mathbf{S}^{(t)} \in \mathbb{R}^{r \times N_t}$ is the unified semantic vector at round t, $\mathbf{U} \in \mathbb{R}^{c \times r}$ is a classifier in multi-class classification. The label can be predicted by logistic regression and calculated by the sigmoid function:

$$\mathbf{Y}^{*(t)} = sigmoid(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\mathbf{U}\mathbf{S}^{(t)}}} \tag{2}$$

Since the true value of the labels $\mathbf{Y}^{(t)}$ are already known, the prediction loss can be expressed by *Log-likelihood Loss*:

$$\begin{aligned}&||\mathbf{U}\mathbf{S}^{(t)}-\mathbf{Y}^{(t)}||_L\\&=-\mathbf{Y}^{(t)}\log(\mathbf{Y}^{*(t)})-(1-\mathbf{Y}^{(t)})\log(1-\mathbf{Y}^{*(t)})\end{aligned}\tag{3}$$

Accordingly, the optimal solution of $\mathbf{S}^{(t)}$ can be approached by gradient descent method. Consequently, the hash code can be quantized as: $\mathbf{B}^{(t)}=\mathrm{sgn}(\mathbf{S}^{(t)})$. The hash code is more discriminative because it strongly correlates to each specific class, whereby the intra-modal semantic similarity and inter-modal semantic similarity are well preserved. In the training stage, the old data $\widetilde{\mathbf{X}}_m^{(t-1)}$ can be generally mapped into their corresponding semantic representation $\widetilde{\mathbf{S}}^{(t-1)}$ by the projection matrix $\mathbf{W}_m$. For online learning, it is expected that the the new data chunk $\mathbf{X}_m^{(t)}$ at each round t can be also mapped into $\mathbf{S}^{(t)}$ by the projection matrix $\mathbf{W}_m$. It is noted that the mappings of $\widetilde{\mathbf{X}}_m^{(t-1)}$ and $\mathbf{X}_m^{(t)}$ are often processed at different stage, and $\mathbf{W}_m$ needs to seek the overall optimal solution between old data and new data chunk. Therefore, the objective function can be formulated as:

$$\min_{\mathbf{W}_m}J_F(\mathbf{W}_m)=||\mathbf{W}_m[\widetilde{\mathbf{X}}_m^{(t-1)},\mathbf{X}_m^{(t)}]-[\widetilde{\mathbf{S}}^{(t-1)},\mathbf{S}^{(t)}]||_F^2\tag{4}$$

where $||\cdot||_F^2$ is the Frobenius norm, $\mathbf{W}_1\in\mathbb{R}^{r\times d_1}$, $\mathbf{W}_2\in\mathbb{R}^{r\times d_2}$ are projection matrices of image and text respectively. By integrating the label consistent item in Eq.(1) and projection item in Eq.(5), the main objective function of the proposed approach, derived at every round t, is defined as:

$$\begin{aligned}&\min_{\mathbf{S}^{(t)},\mathbf{W}_m}J_L(\mathbf{S}^{(t)})+J_F(\mathbf{W}_m)+R(\mathbf{W}_m)\\&=||\mathbf{U}\mathbf{S}^{(t)}-\mathbf{Y}^{(t)}||_L+R(\mathbf{W}_1,\mathbf{W}_2)\\&+\sum_{m=1}^{2}\mu_m||\mathbf{W}_m[\tilde{\mathbf{X}}_m^{(t-1)},\mathbf{X}_m^{(t)}]-[\tilde{\mathbf{S}}^{(t-1)},\mathbf{S}^{(t)}]||_F^2\end{aligned}\tag{5}$$

where $R(\mathbf{W}_m)$ is the regularization term, and μ_m control the nonlinear embedding of different modalities.

3.3. Online Optimization

1) *Initialization*: As shown in Fig. 1, it is necessary to train a suitable $\mathbf{U}$ for initialization, while initializing $\mathbf{W}_1^{(0)},\mathbf{W}_2^{(0)}$ for the modality-specific projections.

Compute $\mathbf{U},\mathbf{W}_1^{(0)},\mathbf{W}_2^{(0)}$: Suppose the first q samples are used to initialize, and $[\mathbf{x}_1^{(j)},\mathbf{x}_1^{(j)}],\mathbf{y}^{(j)}$ are used to denote the image-text feature and label of the jth sample. When focus on solving for $\mathbf{U}$, we can omit the unknown intermediate variable $\mathbf{S}$ in Eq.(6) and then convert the objective formula to:

$$\begin{aligned}\min_{\mathbf{W}_1,\mathbf{W}_2,\mathbf{U}}\sum_{j=1}^{q}&(\alpha||\mathbf{U}\mathbf{W_1}\mathbf{x_1^{(j)}}-\mathbf{y^{(j)}}||_\mathbf{L}\\&+(1-\alpha)||\mathbf{U}\mathbf{W}_2\mathbf{x}_2^{(j)}-\mathbf{y}^{(j)}||_L)\\&+\gamma_1||\mathbf{W}_1||_F^2+\gamma_2||\mathbf{W}_2||_F^2+\gamma_3||\mathbf{U}||_F^2\end{aligned}\tag{6}$$

The predicted label of the j-th sample can be expressed as $\mathbf{y}^{*(j)}=\frac{1}{1+e^{-\mathbf{U}\mathbf{W}_m\mathbf{x}_m^{(j)}}}$, and the *Log-likelihood Loss* is:

$$\begin{aligned}&||\mathbf{U}\mathbf{W}_m\mathbf{x}_m^{(j)}-\mathbf{y}^{(j)}||_L\\&=-\mathbf{y}^{(j)}\log(\mathbf{y}^{*(j)})-(1-\mathbf{y}^{(j)})\log(1-\mathbf{y}^{*(j)})\end{aligned}\tag{7}$$

Then, the solution of $\mathbf{W}_m$ and $\mathbf{U}$ can be obtained:

$$\begin{aligned}\mathbf{W}_1&\leftarrow\mathbf{W}_1-\theta_1\sum_{j=1}^{q}g_{\mathbf{W}_1}^{(j)}\\\mathbf{W}_2&\leftarrow\mathbf{W}_2-\theta_2\sum_{j=1}^{q}g_{\mathbf{W}_2}^{(j)}\\\mathbf{U}&\leftarrow\mathbf{U}-\theta_3\sum_{j=1}^{q}g_{\mathbf{U}}^{(j)}\end{aligned}\tag{8}$$

where θ_1, θ_2,θ_3 denote the learning rates. $g_{\mathbf{W}_1}^{(j)}$, $g_{\mathbf{W}_2}^{(j)}$, $g_{\mathbf{U}}^{(j)}$ denote the gradients. These gradients can be calculated as:

$$\begin{aligned}g_{\mathbf{W}_1}^{(j)}&=\frac{d(||\mathbf{U}\mathbf{W}_1\mathbf{x}_1^{(j)}-\mathbf{y}^{(j)}||_L+\gamma_1||\mathbf{W}_1||_F^2)}{d\mathbf{W}_1}\\&=\frac{d||\mathbf{U}\mathbf{W}_1\mathbf{x}_1^{(j)}-\mathbf{y}^{(j)}||_L}{d\mathbf{y}^{*(j)}}\times\frac{d\mathbf{y}^{*(j)}}{d\mathbf{W}_1}+\gamma_1\mathbf{W}_1\\&=\mathbf{U}^T(\frac{1}{1+e^{-\mathbf{U}\mathbf{W}_1\mathbf{x}_1^{(j)}}}-\mathbf{y}^{(j)})\mathbf{x}_1^{(j)T}+\gamma_1\mathbf{W}_1\end{aligned}\tag{9}$$

$$g_{\mathbf{W}_2}^{(j)}=\mathbf{U}^T(\frac{1}{1+e^{-\mathbf{U}\mathbf{W}_2\mathbf{x}_2^{(j)}}}-\mathbf{y}^{(j)})\mathbf{x}_2^{(j)T}+\gamma_2\mathbf{W}_2\tag{10}$$

$$\begin{aligned}g_{\mathbf{U}}^{(j)}&=\alpha(\frac{1}{1+e^{-\mathbf{U}\mathbf{W}_1\mathbf{x}_1^{(j)}}}-\mathbf{y}^{(j)})\mathbf{x}_1^{(j)T}\mathbf{W}_1^T\\&+(1-\alpha)(\frac{1}{1+e^{-\mathbf{U}\mathbf{W}_2\mathbf{x}_2^{(j)}}}-\mathbf{y}^{(j)})\mathbf{x}_2^{(j)T}\mathbf{W}_2^T+\gamma_3\mathbf{W}_3\end{aligned}\tag{11}$$

2) *Online Update*: In order to achieve higher processing efficiency, we propose a mini-batch online gradient descent (MBOGD) approach to incrementally update the learning variables. At each learning round t, a new data chunk $[\mathbf{X}_1^{(t)},\mathbf{X}_2^{(t)}]$ is added into the training set for online updating, and the detailed updating processes are elaborated below:

Compute $\mathbf{S}^{(t)}$: According to Eq.(6), the loss function with respected to $\mathbf{S}^{(t)}$ is simplified as:

$$\begin{aligned}&\min||\mathbf{U}\mathbf{S}^{(t)}-\mathbf{Y}^{(t)}||_L\\&+\mu_1||\mathbf{W}_1\mathbf{X}_1^{(t)}-\mathbf{S}^{(t)}||_F+\mu_2||\mathbf{W}_2\mathbf{X}_2^{(t)}-\mathbf{S}^{(t)}||_F\end{aligned}\tag{12}$$

Through the analysis in Section 3.2, the optimal solution of $\mathbf{S}^{(t)}$ can be approached by the gradient of Eq.(13):

$$\begin{aligned}\mathbf{S}^{(t)}&=\mathbf{S}^{(t)}-\theta_4 g_{\mathbf{S}}^{(t)}\\&=\mathbf{S}^{(t)}-\theta_4\mathbf{U}^T(\frac{1}{1+e^{-\mathbf{U}\mathbf{S}^{(t)}}}-\mathbf{Y}^{(t)})\\&-\theta_4\mu_1(-\mathbf{W}_1\mathbf{X}_1^{(t)}+\mathbf{S}^{(t)})-\theta_4\mu_2(-\mathbf{W}_2\mathbf{X}_2^{(t)}+\mathbf{S}^{(t)})\end{aligned}\tag{13}$$

where θ_4 is the learning rate.

Compute $\mathbf{B}^{(t)}$: According to Eq.(4), it is easily to derived the hash code of new data pair incrementally at round t:

$$\mathbf{B}^{(t)}=\mathrm{sgn}(\mathbf{S}^{(t)})\tag{14}$$

Next, we can add $\mathbf{B}^{(t)}$ into hash table $\mathbf{B}=[\tilde{\mathbf{B}}^{(t-1)},\mathbf{B}^{(t)}]$.

Update $\mathbf{W}_1$: In Eq.(6), the loss function with respected to $\mathbf{W_1}$ can be simplified as:

$$\min_{\mathbf{W}_1}||\mathbf{W_1}[\widetilde{\mathbf{X}}_1^{(t-1)},\mathbf{X}_1^{(t)}]-[\widetilde{\mathbf{S}}^{(t-1)},\mathbf{S}^{(t)}]||_F^2+R(\mathbf{W}_1)\tag{15}$$

It is noted that the general gradient descent method cannot guarantee the sparse solutions for the semantic representation. Remarkably, sparsity is very important in the processing of high-dimensional data because it reduces the memory and complexity of data representation. Inspired by the findings in

Algorithm 1 Online Label Consistent Hashing

Input: Streaming data chunk $\{[\mathbf{X}_1^{(t)}, \mathbf{X}_2^{(t)}]\}_{t=1}^{N_b}$, label $\{\mathbf{Y}^{(t)}\}_{t=1}^{N_b}$, hash code length r, sample number $q(q \ll N)$ for initialization, parameters γ_1, γ_2, γ_3, μ_1, μ_2 and α;

Output: $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{B}$

1: ***Initialization***: Get first q samples and labels to train classifier $\mathbf{U}$, obtain $\mathbf{W}_1^{(0)}$, $\mathbf{W}_2^{(0)}$, according to Eq.(9)(10)(11)(12);
2: ***Online Updata***: Initialize $\mathbf{U}$, $\mathbf{W}_1 = \mathbf{W}_1^{(0)}$, $\mathbf{W}_2 = \mathbf{W}_2^{(0)}$;
3: **for** $t = 1 \rightarrow N_b$ **do**
4: Compute $\mathbf{S}^{(t)}$ by looping Eq.(14);
5: Compute $\mathbf{B}^{(t)}$ according to Eq.(15);
6: Update $\mathbf{B}$ by $\mathbf{B} = [\mathbf{B}; \mathbf{B}^{(t)}]$;
7: Compute $\eta_1^{(t)}$, $\eta_2^{(t)}$ by $\eta_m^{(t)} = \frac{\eta_m^{(0)}}{\sqrt{t}}$;
8: Update $\mathbf{W}_1$ according to Eqs.(18)(19);
9: Update $\mathbf{W}_2$ according to Eqs.(20)(21);
10: **end for**

work [22], we utilize forward-backward splitting method and *L1 regularization* to ensure the sparse solution of $\mathbf{W}_1$, and the updating processes are formulated as:

$$\begin{aligned}&\mathbf{W}_1^{(t+\frac{1}{2})}=\mathbf{W}_1^{(t)}-\eta_1^{(t)}g_{\mathbf{W}_1}^{(t)}\\&\mathbf{W}_1^{(t+1)}=\arg\min_{\mathbf{W}_1}\left\{\tfrac{1}{2}||\mathbf{W}_1-\mathbf{W}_1^{(t+\frac{1}{2})}||+\eta_1^{t+\frac{1}{2}}\lambda_1||\mathbf{W}_1||_1\right\}\end{aligned} \tag{16}$$

where $\lambda_1 = \frac{\lambda_0}{\sqrt{N}}$, $\eta_1^{(t)} = \frac{\eta_0}{\sqrt{t}}$ is adaptively initialized for updating. The first step is a standard gradient descent step, while the second step can be regarded as the fine-tuning step. The solution appears near the result of gradient descent direction and holds the sparsity. By referring to karush-kuhn-tucker (KKT) condition [22], Eq.(17) can be solved as:

$$\begin{aligned}\mathbf{W}_1 \leftarrow &\operatorname{sgn}(\mathbf{W}_1-\eta_1^{(t)}g_{\mathbf{W}_1}^{(t)})\\&.*\max(0,|\mathbf{W}_1-\eta_1^{(t)}g_{\mathbf{W}_1}^{(t)}|-\eta_1^{(t+\frac{1}{2})}\lambda_1)\end{aligned} \tag{17}$$

$$g_{\mathbf{W}_1}^{(t)} = \frac{d||\mathbf{W}_1\mathbf{X}_1^{(t)}-\mathbf{S}^{(t)}||_F}{d\mathbf{W}_1} = (\mathbf{W}_1\mathbf{X}_1^{(t)}-\mathbf{S}^{(t)})\mathbf{X}_1^{(t)T} \tag{18}$$

Update $\mathbf{W}_2$: Similar to $\mathbf{W}_1$, the solution of $\mathbf{W}_2$ is:

$$\begin{aligned}\mathbf{W}_2 \leftarrow &\operatorname{sgn}(\mathbf{W}_2-\eta_2^{(t)}g_{\mathbf{W}_2}^{(t)})\\&.*\max(0,|\mathbf{W}_2-\eta_2^{(t)}g_{\mathbf{W}_2}^{(t)}|-\eta_2^{(t+\frac{1}{2})}\lambda_2)\end{aligned} \tag{19}$$

$$g_{\mathbf{W}_2}^{(t)} = \frac{d||\mathbf{W}_2\mathbf{X}_2^{(t)}-\mathbf{S}^{(t)}||_F}{d\mathbf{W}_2} = (\mathbf{W}_2\mathbf{X}_2^{(t)}-\mathbf{S}^{(t)})\mathbf{X}_2^{(t)T} \tag{20}$$

3.4. Complexity Analysis

The optimization process of the proposed OLCH framework is shown in Algorithm 1. At each round t, four matrix variables i.e., $\mathbf{S}^{(t)}$, $\mathbf{B}^{(t)}$, $\mathbf{W}_1$, $\mathbf{W}_2$, need to be updated. Since the calculation is only relevant to the current N_t samples, the time complexity is $O(N_t), N_t << N$. Moreover, $\mathbf{U}$, $\mathbf{W}_1$, $\mathbf{W}_2$ are only required to be stored for the updating at the next round, the space complexity is $O(N_t)$.

4. EXPERIMENTS

1) *Datasets*: The popular MIRFlickr [23] and NUS-WIDE [24] datasets are selected for evaluation. The features are selected as the same as in work [12]. For MIRFlickr dataset, 16,738 instances are left whose textual tags appear more than 20 times, and we randomly select 836 pairs as the query set and the rest as training set. For NUS-WIDE dataset, we randomly select 100,000 image-text pairs, and take out 5% of the dataset as the query set and the rest as training set.

2) *Baseline and Parameter*: We compare the proposed OLCH approach with state-of-the-art batch-based methods, i.e., SCM [11], CMFH [4], SMFH [15], FSH [14], SePH [12], IMH [3], DCH [8], GSePH [13], and two online CMH methods, i.e. OCMH [9] and OLSH [10]. The batch-based methods load all the data for training. For online methods, the whole dataset is divided into multiple data chunks to support the evaluation of online performance. The training set of MIRFlickr is split to 15 data chunks of 1,000 sample size except the last one. The training set of NUS-WIDE is split to 9 data chunks of 10,000 sample size except the last one. Further, in order to fairly verify the effectiveness of OLCH in streaming scenarios, CMFH-b, GSePH-b, DCH-b are specially designed as same as in work [20], to enable them to process streaming data blocks. For auxiliary parameters: γ, μ, α, we set $\gamma_1 = \gamma_2 = 10^{-3}$, $\gamma_3 = 1$, $\mu_1 = \mu_2 = 10^{-5}$, $\alpha = 0.3$.

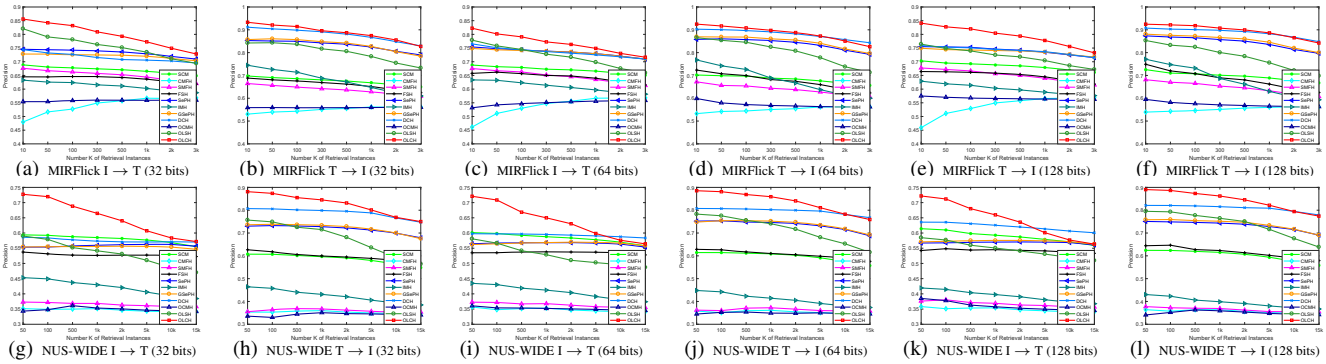
3) *Evaluation Metrics*: the popular mean Average Precision (mAP) and topK-precision [25] are utilized to evaluate the cross-modal retrieval performance. Since similar data samples are often expected to be indexed in the top retrieval list, mAP@100 is selected to evaluate the effectiveness.

4.1. Results and Discussions

1) **Results of Retrieval Accuracy**: In general, online methods may lose retrieval accuracy in comparison with batch-based learning methods, for reason that the online methods often select limited data for training and these data are not reusable during the subsequent learning. Interestingly, as shown in Table 1, it can be observed that: (1) The proposed OLCH yields the best results for all retrieval cases. The main reason lies that the semantic information are well preserved during the hash code learning process, whereby the relevant samples can be successfully indexed. (2) The competing GSePH and DCH methods are able to deliver good performance in offline cases, but which degraded their performances when processing the streaming data. It can be clearly found that the retrieval performances obtained by OLCH are better that than that achieved by GSePH-b and DCH-b methods. As shown in Fig. 2, it can be further found that the topK-precision curves obtained by the proposed OLCH methods have outperformed most baselines, and also performed obviously better than the online methods, i.e., OCMH and OLSH. Further, to verify the advantages of the proposed method in processing the streaming data, Fig. 3 shows the map@100 scores tested on MIRFlick at each round. To be specific,

Table 1: The mAP@100 scores on MIRFlickr and NUS-WIDE datasets

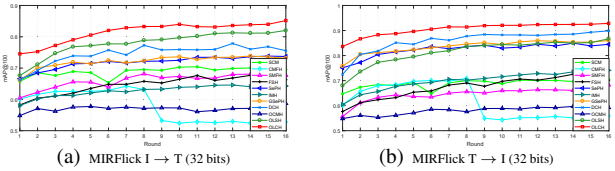
Method	MIRFlickr								NUS-WIDE							
	I → T				T → I				I → T				T → I			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
SCM	0.6943	0.6953	0.6961	0.7099	0.7038	0.7042	0.7120	0.7263	0.5780	0.6126	0.6029	0.6332	0.5312	0.6251	0.6302	0.6401
CMFH	0.5233	0.5299	0.5280	0.5297	0.5553	0.5573	0.5621	0.5659	0.3767	0.3828	0.3874	0.3905	0.3810	0.3880	0.3931	0.3995
SMFH	0.6913	0.6831	0.6830	0.6867	0.6576	0.6711	0.6755	0.6866	0.3831	0.3973	0.4070	0.4033	0.3780	0.3876	0.3853	0.4142
FSH	0.6671	0.6599	0.6751	0.6791	0.7106	0.6957	0.7243	0.7380	0.5353	0.5607	0.5578	0.5637	0.5789	0.6467	0.6494	0.6640
SePH	0.7406	0.7545	0.7606	0.7656	0.8467	0.8574	0.8661	0.8743	0.5595	0.5729	0.5838	0.5853	0.7159	0.7431	0.7648	0.7616
IMH	0.6452	0.6427	0.6465	0.6370	0.7324	0.7412	0.7568	0.7636	0.4967	0.4776	0.4608	0.4476	0.5024	0.4891	0.4747	0.4581
GSePH	0.7374	0.7388	0.7551	0.7563	0.8598	0.8649	0.8745	0.8823	0.5583	0.5724	0.5819	0.5890	0.7335	0.7467	0.7613	0.7703
DCH	0.7476	0.7546	0.7825	0.7632	0.9025	0.9117	0.9078	0.9070	0.6128	0.6088	0.6091	0.6453	0.8090	0.8172	0.8101	0.8239
CMFH-b	0.5905	0.5917	0.6023	0.5990	0.5797	0.5850	0.5939	0.6010	0.4099	0.4345	0.4441	0.4567	0.4100	0.4332	0.4481	0.4526
GSePH-b	0.6741	0.6604	0.6847	0.6986	0.7417	0.7575	0.7843	0.7859	0.5206	0.5293	0.5346	0.5445	0.6492	0.6826	0.7053	0.7092
DCH-b	0.5353	0.5640	0.5874	0.5983	0.6200	0.6150	0.6142	0.6065	0.3732	0.4863	0.5415	0.5004	0.6982	0.8015	0.7951	0.7270
OCMH	0.5746	0.5749	0.5602	0.5889	0.5796	0.5792	0.6001	0.6021	0.4784	0.4733	0.4178	0.4334	0.4920	0.5144	0.4331	0.4704
OLSH	0.7973	0.8136	0.7882	0.7981	0.7890	0.8587	0.8693	0.8521	0.6799	0.6897	0.6857	0.7011	0.8282	0.8312	0.8556	0.8465
OLCH-f	0.7992	0.8088	0.8100	0.8237	0.8639	0.8820	0.9173	0.9231	0.6681	0.6760	0.6928	0.6900	0. 8371	0.8547	0.8789	0.8654
OLCH-m	**0.8356**	0.8398	0.8469	0.8481	0.8836	**0.9225**	0.9278	**0.9301**	0.7122	0.7341	**0.7456**	0.7399	**0.8645**	0. 8799	0.8823	**0.8946**
OLCH	0.8222	**0.8433**	**0.8506**	**0.8576**	**0.8937**	0.9216	**0.9313**	0.9228	**0.7172**	**0.7363**	0.7402	**0.7406**	0.8601	**0.8810**	**0.8917**	0.8926



(a) MIRFlick I → T (32 bits) (b) MIRFlick T → I (32 bits) (c) MIRFlick I → T (64 bits) (d) MIRFlick T → I (64 bits) (e) MIRFlick T → I (128 bits) (f) MIRFlick T → I (128 bits)

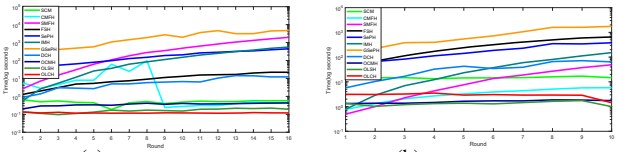
(g) NUS-WIDE I → T (32 bits) (h) NUS-WIDE T → I (32 bits) (i) NUS-WIDE I → T (64 bits) (j) NUS-WIDE T → I (64 bits) (k) NUS-WIDE I → T (128 bits) (l) NUS-WIDE T → I (128 bits)

Fig. 2: The representative topK-precision curves tested on MIRFlick and NUS-WIDE.



(a) MIRFlick I → T (32 bits) (b) MIRFlick T → I (32 bits)

Fig. 3: The mAP@100 tested on MIRFlick at each round.



(a) MIRFlick (32 bits) (b) NUS-WIDE (32 bits)

Fig. 4: The training time at each round.

the batch-based methods retrain hash functions on the whole training dataset when a new data chunk arrives, while only the new data is utilized to update the hash function for online learning methods. It can be observed that the growth curves obtained by SCM, CMFH, SMFH and DCH all appear obvious fluctuations, which indicates that these batch-based methods are unstable to processing the streaming data. By contrast, the proposed OLCH performs better at each round and yields a more stable and consistent growth curve. That is, our proposed method has achieved stable and outstanding performance in processing the streaming data.

2) **Result of Training Time**: Fig. 4 shows the training time of all compared methods tested on MIRFlickr and NUS-WIDE dataset, and x-coordinate indicates the increasing of learning round. Since the training times of batch-based methods are much larger than that obtained by online methods, the figures show the log value of seconds to represent the y-coordinate. It can be found that the most significant observation is that the time cost of OLCH does not increase at each round, which proves the time complexity of OLCH is only related to the new arriving data. Table 2 shows the total training time related to all training data, the proposed OLCH method performs sufficient fast in processing large-scale dataset.

Table 2: The total time of all training data, which is the sum of time of all round for online methods, and the training time of loading all data for batch-based methods.

Method	MIRFlickr				NUS-WIDE			
	16	32	64	128	16	32	64	128
SMFH	470.19	460.53	423.26	457.07	58.67	53.81	48.11	58.98
FSH	24.38	23.56	24.61	27.69	663.35	674.09	698.43	699.74
SePH	960.02	1125.51	1468.58	2360.32	286.48	377.17	565.64	818.09
IMH	507.56	486.50	479.17	495.23	176.55	180.37	176.47	180.37
GSePH	10438.77	11464.38	31158.06	58250.84	1055.83	1832.51	3151.10	5610.91
DCH	8.55	14.04	49.24	539.39	32.01	74.22	316.99	1344.30
OCMH	5.04	5.77	7.57	11.52	13.34	16.17	19.66	27.90
OLSH	2.11	2.63	5.60	16.62	7.90	13.95	25.45	78.36
OLCH-m	182.33	215.33	259.70	324.56	1185.74	1340.53	2632.09	6274.04
OLCH	1.66	1.92	2.35	3.31	28.15	29.04	33.62	41.97

3) **Result of Ablation Studies**: The OLCH-m represents the OLCH without MBOGD, the OLCH-f represents the OLCH without forward-backward splitting and *L1 regularization*. Table 1 shows the mAP@100 scores tested by

different learning mechanisms, it can be seen that OLCH and OLCH-m deliver the similar results, while performing better than OLCH-f. The main reason lies that forward-backward splitting and *L1 regularization* are able to deliver discriminative hashing projection functions, whereby the discriminative hash codes are obtained for high retrieval performance. Comparing with OLCH-m, as shown in Table 2, OLCH significantly speeds up the learning process. That is, the proposed OLCH runs very fast with high retrieval performance. The experiments have shown its outstanding performance.

5. CONCLUSION

This paper has proposed an efficient online label consistent hashing method for cross-modal retrieval. Specifically, the designed online semantic representation learning framework is able to incrementally learn hash codes for the current arriving data, while adaptively updating the hash functions at a streaming manner. Accordingly, the discriminative hash codes are adaptively learned online with the high discriminative capability, and the experimental results show its outstanding performance in comparison with the-state-of-arts.

6. REFERENCES

[1] J.M.Wang Y.Cao, M.S.Long and H.Zhu, "Correlation autoencoder hashing for supervised cross-modal search," in *ACM ICMR*, 2016, pp. 197–204.

[2] C.X.Li Y.L.Shi S.Q.Guo H.J.Huang, R.Yang and X.S.Xu, "Supervised cross-modal hashing without relaxation," in *IEEE ICME*, 2017, pp. 1159–1164.

[3] Y.Yang Z.Huang J.K.Song, Y.Yang and H.T.Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *ACM SIGMOD*, 2013, pp. 785–796.

[4] Y.C.Guo G.G.Ding and J.L.Zhou, "Collective matrix factorization hashing for multimodal data," in *IEEE CVPR*, 2014, pp. 2075–2082.

[5] Y.Zhen B.Liu Y.Wei, Y.Q.Song and Q.Yang, "Heterogeneous translated hashing: A scalable solution towards multi-modal similarity search," *ACM TKDD*, vol. 10, no. 4, pp. 36, 2016.

[6] R.C.Hong L.Zhang, Y.D.Zhang and Q.Tian, "Full-space local topology extraction for cross-modal retrieval," *IEEE TTP*, vol. 24, no. 7, pp. 2212–2224, 2015.

[7] Q.Y.Jiang and W.J.Li, "Deep cross-modal hashing," in *IEEE CVPR*, 2017, pp. 3232–3240.

[8] Y.Yang H.T.Shen X.Xu, F.M.Shen and X.L.Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE TIP*, vol. 26, no. 5, pp. 2494–2507, 2017.

[9] L.Xie, J.L.Shen, and L.Zhu, "Online cross-modal hashing for web image retrieval," in *AAAI*, 2016, pp. 294–300.

[10] L.S.Yan X.W.Kong Q.T.Su C.M.Zhang T.Yao, G.Wang and Q.Tian, "Online latent semantic hashing for cross-media retrieval," *PR*, vol. 89, pp. 1–11, 2019.

[11] D.Q.Zhang and W.J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *AAAI*, 2014, pp. 2177–2183.

[12] M.Q.Hu Z.J.Lin, G.G.Ding and J.M.Wang, "Semantics-preserving hashing for cross-view retrieval," in *IEEE CVPR*, 2015, pp. 3864–3872.

[13] D.Mandal, K.N.Chaudhury, and S.Biswas, "Generalized semantic preserving hashing for n-label cross-modal retrieval," in *IEEE CVPR*, 2017, pp. 4076–4084.

[14] Y.J.Wu F.Y.Huang H.Liu, R.R.Ji and B.C.Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *IEEE CVPR*, 2017, pp. 7380–7388.

[15] K.Wang J.Tang and L.Shao, "Supervised matrix factorization hashing for cross-modal retrieval," *IEEE TIP*, vol. 25, no. 7, pp. 3157–3166, 2016.

[16] E.Yu J.D.Sun L.Wang, L.Zhu and H.X.Zhang, "Fusion-supervised deep cross-modal hashing," in *IEEE ICME*, 2019, pp. 37–42.

[17] F.Cakir and S.Sclaroff, "Adaptive hashing for fast similarity search," in *IEEE ICCV*, 2015, pp. 1044–1052.

[18] S.A.Bargal F.Cakir and S.Sclaroff, "Online supervised hashing," *Computer Vision and Image Understanding*, vol. 156, pp. 162–173, 2017.

[19] H.Liu M.B.Lin, R.R.Ji and Y.J.Wu, "Supervised online hashing via hadamard codebook learning," in *ACM MM*, 2018, pp. 1635–1643.

[20] Y.Q.An X.B.Gao D.Wang, Q.Wang and Y.M.Tian, "Online collective matrix factorization hashing for large-scale cross-media retrieval," in *ACM SIGIR*, 2020, pp. 1409–1418.

[21] T. Wang, Y. Lu, J. Wang, H. N. Dai, X. Zheng, and W. Jia, "Eihdp: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for iot systems," *IEEE Transactions on Computers*, 2021.

[22] J.Duchi and Y.Singer, "Efficient online and batch learning using forward backward splitting," *Journal of Machine Learning Research*, vol. 10, no. 18, pp. 2899–2934, 2009.

[23] M.J.Huiskes and M.S.Lew, "The mir flickr retrieval evaluation," in *ACM MIR*, 2008, pp. 39–43.

[24] R.C.Hong H.J.Li T.Chua, J.H.Tang and Z.P.Luo, "Nus-wide: a real-world web image database from national university of singapore," in *ACM CIVR*, 2009, pp. 1–9.

[25] X. Liu, Z. Hu, H. Ling, and Y. M. Cheung, "Mtfh: A matrix tri-factorization hashing framework for efficient cross-modal retrieval," *IEEE TPAMI*, vol. 43, no. 3, pp. 964–981, 2021.