2-2016

Online cross-modal hashing for web image retrieval

Liang XIE

Jialie SHEN
Singapore Management University, jlshen@smu.edu.sg

Lei ZHU

Published in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16): Phoenix, AZ, February 12-17, 2016. pp. 294-300.
http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12125

Online Cross-Modal Hashing for Web Image Retrieval

Liang Xie
Department of Mathematics
Wuhan University of Technology, China
whutxl@hotmail.com

Jialie Shen * and **Lei Zhu**
School of Information Systems
Singapore Management University, Singapore
jlshen@smu.edu.sg, leizhu0608@gmail.com

Abstract

Cross-modal hashing (CMH) is an efficient technique for the fast retrieval of web image data, and it has gained a lot of attentions recently. However, traditional CMH methods usually apply batch learning for generating hash functions and codes. They are inefficient for the retrieval of web images which usually have streaming fashion. Online learning can be exploited for CMH. But existing online hashing methods still cannot solve two essential problems: efficient updating of hash codes and analysis of cross-modal correlation. In this paper, we propose Online Cross-modal Hashing (OCMH) which can effectively address the above two problems by learning the shared latent codes (SLC). In OCMH, hash codes can be represented by the permanent SLC and dynamic transfer matrix. Therefore, inefficient updating of hash codes is transformed to the efficient updating of SLC and transfer matrix, and the time complexity is irrelevant to the database size. Moreover, SLC is shared by all the modalities, and thus it can encode the latent cross-modal correlation, which further improves the overall cross-modal correlation between heterogeneous data. Experimental results on two real-world multi-modal web image datasets: MIR Flickr and NUS-WIDE, demonstrate the effectiveness and efficiency of OCMH for online cross-modal web image retrieval.

Introduction

Web image is typical multi-modal data, which consists of multiple information type, such as visual contents and text tags. Cross-modal hashing (CMH) (Song et al. 2013; Zhang and Li 2014; Xie et al. 2015) has gained a lot of attentions for the fast retrieval of web image data. CMH combines the advantages of cross-modal analysis (Costa Pereira et al. 2014; Zhai, Peng, and Xiao 2013; Xie, Pan, and Lu 2015) and hashing technology (Weiss, Torralba, and Fergus 2009; Zhang et al. 2010; Zhu, Shen, and Xie 2015), it can efficiently solve the retrieval of heterogenous modalities.

The basic idea of CMH methods is to project the information from different modalities into a unified hash space, where hamming distance can be applied as the metric to measure distance. They often learn hash functions in offline process by batch learning. Then the hash codes of all database data are computed by the learned hash functions. As a result, existing CMH methods might not be able to achieve good performance under environment of online real-world web image retrieval, where they generally ignore that data usually arrive in streaming fashion. The online images (e.g., those on Flickr and Google), are rapidly increased as time goes on. For example, millions of new images are uploaded to Flickr by users each day. If new images are added to the web database, existing CMH methods have to accumulate all the database data to retrain new hash functions, and recomputed the hash codes of whole database. They are obviously inefficient in the learning process of hash functions and codes, especially when the database is frequently updated. As an emerging technology, online hashing technique (Huang, Yang, and Zheng 2013), can be applied to cope with the online retrieval of streaming database. However, existing online hashing methods cannot be directly applied to cross-modal retrieval of web images, in that they have ignored two essential problems:

- Existing online hashing methods are inefficient when updating hash codes. They only focus on the online learning of hash functions, but ignore efficient updating of hash codes. In general online hashing process, the hash function can be efficiently retrained when new data arrive. However, the change of hash functions will result in the change of hamming space. In order to make the new data and old data to be effectively matched, the whole database should be accumulated to compute their new hash codes by the updated hash functions. As a result, the time complexity of updating hash codes depends on the size of whole database, and it is obviously very inefficient in the online scenario. For an efficient online hashing method, the updating time of hash codes must be irrelevant to database size.

- The cross-modal correlation is not analyzed by online hashing. Cross-modal correlation describes the relationship between different modalities, thus it plays an important role in cross-modal retrieval. Due to the well-known semantic gap between different modalities, cross-modal correlation is difficult to be analyzed. Moreover, with the change of database, the correlation between heterogeneous data is also varied. The analysis of ever-changing cross-modal correlation poses great challenge to online hashing method.

*Corresponding Author

In this paper, we propose Online Cross-Modal Hashing (OCMH) for the fast retrieval of streaming web images. To address two aforementioned issues, OCMH decomposes hash codes of all modalities to shared latent codes (SLC) and transfer matrix. In the online learning process, SLC can be incrementally updated by preserving its old codes, and the improved information from new data can be preserved by the dynamic transfer matrix. As a result, inefficient updating of hash codes is transformed to efficient updating of SLC. Moreover, SLC is shared by different modalities, thus it can encode the latent cross-modal correlation, which are combined with the basic cross-modal correlation in OCMH. Therefore, OCMH can thoroughly analyze the cross-modal correlation in the online learning process. The contributions of this paper are listed as follows:

- We propose OCMH which is efficient in the online scenario where images arrive in streaming fashion. Unlike previous online hashing schemes which cannot efficiently update hash codes, OCMH can update them online by optimizing the SLC and transfer matrix.
- OCMH specially considers cross-modal correlation in the online learning process. SLC can encode the latent cross-modal correlation, which improves the cross-modal analysis of OCMH. As a result, OCMH can effectively solve the retrieval of heterogeneous modalities, while it also ensures the learning efficiency.
- Experimental results demonstrate both the effectiveness and efficiency of OCMH compared to other cross-modal hashing and online hashing methods.

Related Work

Cross-Modal Hashing

In recent years, many efforts have been devoted to cross-modal hashing (CMH). Most CMH methods focus on analyzing the cross-modal correlation between heterogenous data by cross-modal/multi-modal techniques (Zhu et al. 2015). Cross-View Hashing (CVH) (Kumar and Udupa 2011) correlates different modalities by learning similar hash codes for them. Inter-Media Hashing (IMH) (Song et al. 2013) models the cross-modal correlation by inter-media and intra-media consistency. Latent Semantic Sparse Hashing (LSSH) (Zhou, Ding, and Guo 2014) first learns latent semantic features for images and texts respectively, then the learned latent features are correlated in a unified hash space. Collective Matrix Factorization Hashing (CMFH) (Ding, Guo, and Zhou 2014) uses Collective Matrix Factorization (Singh and Gordon 2008) to obtain joint hash codes which can correlate different modalities. Some CMH methods also consider the quantization effect of hash codes for multi-modal data. Semantic Correlation Maximization (SCM) (Zhang and Li 2014) adopts the sequential learning (Wang, Kumar, and Chang 2012) to improve the performance of hash codes. Quantized Correlation Hashing (QCH) (Wu et al. 2015) takes into consideration the quantization loss for cross-modal correlation.

Due to the advantages of hashing technology, CMH methods are efficient in the search process. However, most of them are not efficient in the learning process. SCM and Linear Cross-Modal Hashing (LCMH) (Zhu et al. 2013) improve learning process with linear time complexity. But they are still batch learning based methods, which are not suited to the online scenario. To the best of our knowledge, currently there are no CMH methods which use online learning for hash functions or codes.

Table 1: Comparison of Time Complexity ($N_t \ll N$).

Method	CMH	Online Hashing	OCMH
Hash Code Learning	$O(N)$	$O(N)$	$O(N_t)$
Hash Function Learning	$O(N)$	$O(N_t)$	$O(N_t)$

Online Hashing

Online Hashing, which exploits online learning (Liberty 2013) for hashing process, is practical in the real-world applications, but there are not much studies about it so far (Wang et al. 2014). The study of (Jain et al. 2009) may be the first attempt to use online learning for hashing, it first designs an online metric learning algorithm, then it updates the changed hash codes. However, the search time of changed hash codes depends on the size of whole database. Online Kernel-based Hashing (OKH) (Huang, Yang, and Zheng 2013) and Online Sketching Hashing (OSH) (Leng et al. 2015) both learn hash functions in online process. However, they are not able to update hash codes online. At each round of updating, since the hash functions are changed, they have to accumulate all database data to recompute the hash codes, which is obviously inefficient.

Brief Comparison

Table 1 makes a comparison about learning time of the three types of hashing technique, where N is the database size, N_t is the new data size and $N_t \ll N$. Table 1 demonstrates that OCMH is the most efficient, online hashing (e.g. OSH) is partly efficient and CMH (e.g. CVH) is inefficient for online web image retrieval.

Method of Online Cross-Modal Hashing

The graphical illustration of OCMH is shown in Fig.1. Hash codes of each modality are constructed from SLC H and variable matrix V_m. Then inefficient updating of hashing codes is transformed to online learning of H and V_m. In the online learning process, the old part of H is permanent, only new codes are added to H.

Problem Description

Suppose the database consists of streaming multi-modal documents, each of them is an image-text pair. At each round t, new data chunk $X^{(t)} = [X_1^{(t)}, X_2^{(t)}]$ is added to the database, where $X_1^{(t)} \in R^{N_t \times d_1}$ and $X_2^{(t)} \in R^{N_t \times d_2}$ denote the feature matrices of image and text respectively, N_t is the size of new data. There also exist old data $\widetilde{X} = [\widetilde{X}_1, \widetilde{X}_2]$ in

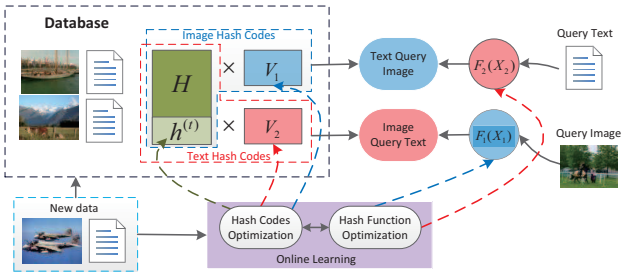


Figure 1: The graphical illustration of OCMH.

database. Then the total database contains $X = [X_1, X_2]$, where $X_m = [\widetilde{X}_m^T, (X_m^{(t)})^T]^T$ and $m = 1, 2$.

At each round t, we aim to learn hash function $f_m(x_m)$ and database hash codes F_m for each modality m. The hash function is defined as:

$$f_m(x_m) = \text{sgn}\left(x_m W_m + b_m\right) \tag{1}$$

where x_m is the feature vector of modality m. If we set $x_m = [x_m, 1]$, $W_m = [W_m^T, b_m^T]^T$. Then the hash function can be reformulated as $f_m(x_m) = \text{sgn}\left(x_m W_m\right)$. Hash functions from different modalities project all data into a unified hash space, and the database hash codes of each modality can be obtained as $F_m = \text{sgn}\left(X_m W_m\right)$.

Suppose we have learned $\widetilde{W}_m$ and hash codes $\widetilde{F}_m = \widetilde{X}_m \widetilde{W}_m$ at previous round $t-1$. At round t we should first learn new hash weights W_m online. Then with the updating of hash weights, if we directly update hash codes, we have to accumulate all database data. This updating process of hash codes is obviously very inefficient, especially when new data are added frequently. Therefore, one major goal of our OCMH is to efficiently update hash codes online, which will be solved in following subsections.

Formulation

At first we optimize the basic cross-modal correlation between images and texts. The hamming distance between image and text hash codes in a pair should be minimal, thus we obtain the following objective function:

$$min \left\|X_1 W_1 - X_2 W_2\right\|_F^2 \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The non-differentiable $\text{sgn}(\cdot)$ is dropped to avoid the NP-hard solution (Kumar and Udupa 2011).

The optimization of W_m will result in the updating of whole hash codes F_m. In ideal case, we may want to preserve existing hash codes $\widetilde{F}_m$, and only compute codes of new data to incrementally obtain F_m. However, $\widetilde{F}_m$ is only related to $\widetilde{W}_m$ which is learned at previous round. It is obviously that $\widetilde{F}_m$ cannot match new hash codes which are computed by W_m. In order to efficiently updating F_m, we assume that it is constructed from a permanent SLC H and a dynamic transfer matrix V_m. Then the inefficient updating of F_m is transformed to efficient updating of H. The decomposition of F_m can be represented as the following constraints:

$$X_m W_m = H V_m \quad (m = 1, 2) \tag{3}$$

where H consists of old latent codes $\widetilde{H}$ and new latent codes $H^{(t)}$, that is $H = [\widetilde{H}^T, (H^{(t)})^T]^T$.

At each round t, V_m is updated, and H is only added with codes of new data $H^{(t)}$. The size of V_m is irrelevant to the database size N, and is much smaller than the total hash codes. As a result, instead of updating whole F_m, we only need to update V_m, which is very efficient. Moreover, H is shared by all modalities, it encodes the latent cross-modal correlation which can effectively supplement the basic cross-modal correlation described above.

By combining the objective function Eq.(2) with constraints Eq.(3), we can arrive at the following formulation for learning hash functions and SLC:

$$\begin{aligned} \min \quad & \left\|X_1 W_1 - X_2 W_2\right\|_F^2 + \lambda \sum_{m=1}^{2} \theta_m \left\|X_m W_m - H V_m\right\|_F^2 \\ & + \lambda \left(\alpha \left\|H\right\|_F^2 + \beta \sum_{m=1}^{2} \theta_m \left\|V_m\right\|_F^2\right) \end{aligned} \tag{4}$$

where λ, α and β are regularization parameters. θ_m denotes the importance of image and text, and $\theta_1 + \theta_2 = 1$.

Online Optimization

In this section we propose an online optimization of W_m and H from Eq.(4). The objective function is convex to each W_m or H, thus we can use the alternative process for optimization.

At first we consider the optimization of SLC H and transfer matrix V_m. By setting the derivative of Eq.(4) w.r.t V_m to zeros, we can easily update V_m by:

$$V_m = \left(C_H + \beta I\right)^{-1} E_m W_m \tag{5}$$

where $C_H = H^T H$ and $E_m = H^T X_m$.

According to Theorem 1, we can easily updated V_m in an online process:

Theorem 1. *The time complexity of computing C_H and E_m is $O(N_t)$, which is linear to the size of new data.*

Proof. We can easily obtain the following equation:

$$\begin{aligned} C_H &= \left[\widetilde{H}^T, \left(H^{(t)}\right)^T\right] \begin{bmatrix} \widetilde{H} \\ H^{(t)} \end{bmatrix} \\ &= \widetilde{H}^T \widetilde{H} + \left(H^{(t)}\right)^T H^{(t)} = \widetilde{C}_H + \left(H^{(t)}\right)^T H^{(t)} \end{aligned} \tag{6}$$

$\widetilde{C}_H = \widetilde{H}^T \widetilde{H}$ is related to old data, it is computed at the previous round, so we only need to compute $\left(H^{(t)}\right)^T H^{(t)}$. We can omit the code length and feature dimensions which are irrelevant to the data size, then the computing time complexity of $\widetilde{C}_H$ is $O(N_t)$. Similarly, we can obtain:

$$E_m = \widetilde{H}^T \widetilde{X}_m + \left(H^{(t)}\right)^T X_m^{(t)} = \widetilde{E}_m + \left(H^{(t)}\right)^T X_m^{(t)} \tag{7}$$

Then the computation time of E_m is also linear to new data size N_t. □

From Theorem 1, we can easily obtain that the updating time of V_m is $O(N_t)$.

Since $\widetilde{H}$ is permanent, we only need to update $H^{(t)}$, which is computed by:

$$H^{(t)} = \sum_{m=1}^{2} \theta_m X_m^{(t)} W_m V_m^T \left(\alpha I + \sum_{m=1}^{2} \theta_m V_m V_m^T \right)^{-1} \quad (8)$$

According to Eq.(8), the time of computing $H^{(t)}$ is also $O(N_t)$, and it is irrelevant to the database size N. Then H can be efficiently updated.

At last we consider the updating of W_m. Substituting Eq.(5) to Eq.(4), then Eq.(4) can be transformed to:

$$\begin{aligned} min \quad & Tr\left(\sum_{m=1}^{M} W_m^T \left((\lambda\theta_m + 1) C_m - \lambda\theta_m B_m \right) W_m \right) \\ & - Tr\left(W_1^T C_{12} W_2 + W_2^T C_{21} W_1 \right) \end{aligned} \quad (9)$$

where $Tr(\cdot)$ denotes the trace operator, and :

$$\begin{aligned} & C_m = X_m^T X_m, \; C_{mn} = X_m^T X_n \;\; (m, n = 1, 2) \\ & B_m = E_m^T (C_H + \beta I)^{-1} E_m \end{aligned} \quad (10)$$

By setting the derivative of W_2 w.r.t Eq.(9) to zero, we can obtain:

$$W_2 = \left((\lambda\theta_2 + 1) C_2 - \lambda\theta_2 B_2 \right)^{-1} C_{21} W_1 \quad (11)$$

In order to avoid the trivial solution, we add a constraint for W_1. By substituting Eq.(11) into Eq.(9), we can obtain W_1 by solving the following eigenvalue problem:

$$\begin{aligned} max \quad & Tr\left(W_1^T C_{12} ((\lambda\theta_2 + 1) C_2 - \lambda\theta_2 B_2)^{-1} C_{21} W_1 \right) \\ s.t. \quad & W_1^T \left((\lambda\theta_1 + 1) C_1 - \lambda\theta_1 B_1 \right) W_1 = I \end{aligned} \quad (12)$$

We can also efficiently compute Eq.(12) and Eq.(11), according to Theorem 2:

Theorem 2. *For $n, m = 1, 2$, the time complexity of computing C_m and C_{mn} is $O(N_t)$, which is linear to the size of new data.*

Proof. Similar to Theorem 1, C_m and C_{mn} can be computed by:

$$C_m = \widetilde{C}_m + \left(X_m^{(t)} \right)^T X_m^{(t)}, C_{mn} = \widetilde{C}_{mn} + \left(X_m^{(t)} \right)^T X_n^{(t)} \quad (13)$$

where $\widetilde{C}_m = \widetilde{X}_m^T \widetilde{X}_m$ and $\widetilde{C}_{mn} = \widetilde{X}_m^T \widetilde{X}_n$ has been computed at previous round, thus we only need to compute new data, and the time complexity is $O(N_t)$. □

The updating of Eq.(11) is based on the computation of C_m and C_{mn}, thus W_2 can be efficiently computed with time complexity $O(N_t)$. Moreover, in Eq.(12), we only need to solve the eigenvectors of a $d_1 \times d_1$ matrix, which is obviously irrelevant to both database size N and new data size N_t. As a result, the computation of all W_m will cost $O(N_t)$.

The whole optimization process at each round t is presented in Algorithm 1, where T_{iter} denotes the total numbers of iterations. At each round, since V_m has been optimized by old data, we do not need much iterations for updating V_m, as well as the updating of H. Thus we set $T_{iter} = 3$ in our implementations.

Algorithm 1 Optimizing algorithm at round t

Input:

$X_m^{(t)}$, V_m, $\widetilde{H}$, $\widetilde{C}_H$, $\widetilde{E}_m$, $\widetilde{C}_m$, $\widetilde{C}_{mn}$ (m, n=1,2)

Output:

W_m, H, C_H, E_m, C_m, C_{mn},V_m

Update C_m, C_{mn} according to Eq.(10);
Initialize $H^{(t)}$ randomly;
for $iter < T_{iter}$ **do**
  Compute W_1 by solving the eigenvalue problem of Eq.(12);
  Compute W_2 according to Eq.(11);
  Update C_H and E_m according to Eq.(6) and Eq.(7);
  Update V_m according to Eq.(5);
  Compute $H^{(t)}$ according to Eq.(8);
end for
Update H by $H = [H^T, \mathrm{sgn}(H^{(t)})^T]^T$;

Algorithm 1 is efficient in both time and memory cost. At each round, the matrix C_H, E_m, C_m, C_{mn},V_m is preserved for the updating at next round. The size of these matrices are only related to feature dimensions and hash code length, thus they are very small and occupy not much memory space. Moreover, we have discussed that the updating of W_m, V_m and H is linear to new data size. Since T_{iter} is small, the time complexity of whole optimizing algorithm is $O(N_t)$, which is linear to new data size. As a result, we can expect a stable learning time which is irrelevant to database size at any rounds.

Retrieval Process

After the updating of H, the new hash codes of each modality can be represented by HV_m. We consider two types of cross-modal retrieval tasks. The first is image query, where image example is used as query to search texts in database. The other is text query, where text example is used to search images in database. Suppose the new query is a text x_2, we can compute its hash codes by $x_2 W_2$, and the distance vector between query and database images are $dist = x_2 W_2 V_1^T H^T$. We can approximate the distance by $\mathrm{sgn}\left(x_2 W_2 V_1^T\right) H^T$, then the distance between hash codes are transferred to distance between query latent codes and SLC. The latent codes of each query can be computed as:

$$h_q(x_m) = \mathrm{sgn}\left(x_m W_m V_n^T\right), \; m, n = 1, 2 \; and \; m \neq n \quad (14)$$

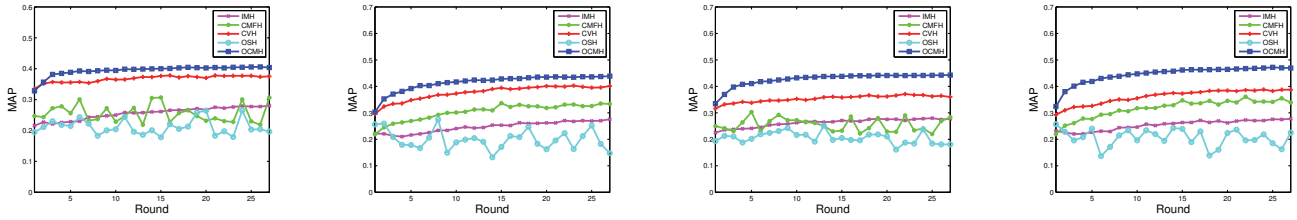
In the retrieval process, given a query from any modality, we compute the hamming distance between $h_q(x_m)$ and H to search relevant data.

Experiments

Datasets and Features

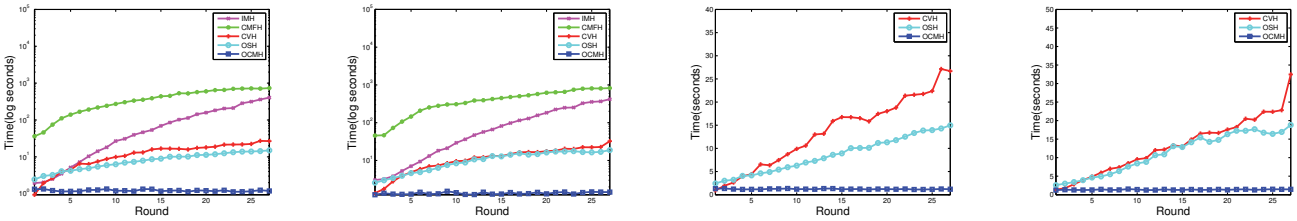
In our experiments, two real-world web image datasets: MIR Flickr and NUS-WIDE, are used to evaluate the effectiveness and efficiency of OCMH.

MIR Flickr consists of 25,000 images downloaded from Flickr. Each image is associated with several text words, thus



(a) Image query (32 bits) (b) Text query (32 bits) (c) Image query (64 bits) (d) Text query (64 bits)

Figure 2: The MAP scores of NUS-WIDE at each round, with 32 and 64 bits of hash codes.



(a) 32 bits (b) 64 bits (c) 32 bits (d) 64 bits

Figure 3: Time comparison on NUS-WIDE with 32 and 64 bits, both log seconds and seconds are used.

they can be considered as a multi-modal image-text pair. Images are annotated with 38 class labels which are used as the ground truth. In the retrieval, images which share at least one same label are considered as relevant. We randomly select 1,000 images and their associated text as queries. To support the evaluation of online performance, the whole dataset is split to 13 data chunks, each of the first 12 chunks contains 2,000 pairs, and the last chunk contains 1,000 pairs.

NUS-WIDE contains 269,648 image-text pairs which are also downloaded from Flickr, each pair is labeled by 81 concepts that can be used for evaluation. We randomly select 2,000 images and associated texts as queries. The whole dataset is split to 27 chunks, each of the first 26 chunks contains 10,000 pairs, and the last chunk contains 9,648 pairs.

On MIR Flickr, we directly use the image and text features provided in (Guillaumin, Verbeek, and Schmid 2010), including 15 image features and one binary text feature. We also directly use 6 image features and one binary text feature provided by (Chua et al. 2009). Since the dimensions of image feature are too large, we use Kernel PCA (KPCA) (Schölkopf, Smola, and Müller 1997) to combine image features and reduce their dimensions. Finally we obtain 500-D visual feature for images on MIR Flickr, and 100-D visual feature for images on NUS-WIDE.

Experimental Settings

In the implementation of OCMH, we set the regularization parameters λ, α and β to 10^{-6}. Since text usually contains more semantic information than image, we set $\theta_1 = 0.3$ and $\theta_2 = 0.7$.

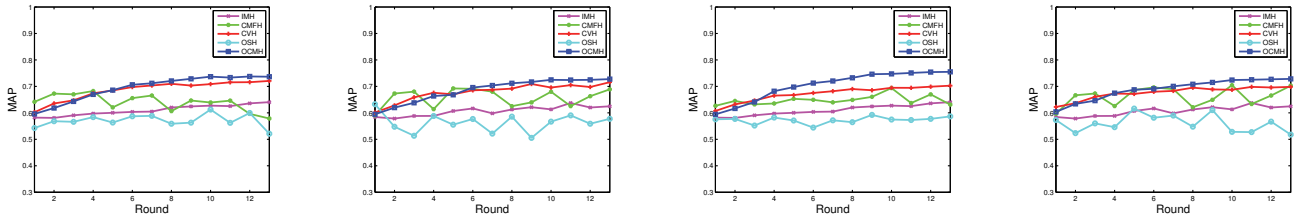
There are no existing similar hashing methods to OCMH, so we use CMH and online hashing methods for comparison. We compare our OCMH to three representative cross-modal hashing methods, including Cross-View Hashing (CVH) (Kumar and Udupa 2011), Inter-Media Hashing (IMH) (Song et al. 2013) and Collective Matrix Factorization Hashing (CMFH) (Ding, Guo, and Zhou 2014). All of them use batch learning for hash functions and codes, thus we have to retrain them at each round. We also compare OCMH to Online Sketching Hashing (OSH) (Leng et al. 2015) which is a uni-modal online hashing method.

Mean average precision (MAP) (Song et al. 2013) is used to measure the effect of retrieval, and MAP scores are computed on the top 50 retrieved documents of each query. Moreover, we evaluate the learning time of all methods to measure the efficiency of retrieval, and learning time is the total time of learning hash functions and updating hash codes at each round. All the experiments are conducted on a computer with Intel Core(TM) i5 2.6GHz 2 processors and 12.0GB RAM.

Results of Cross-Modal Retrieval

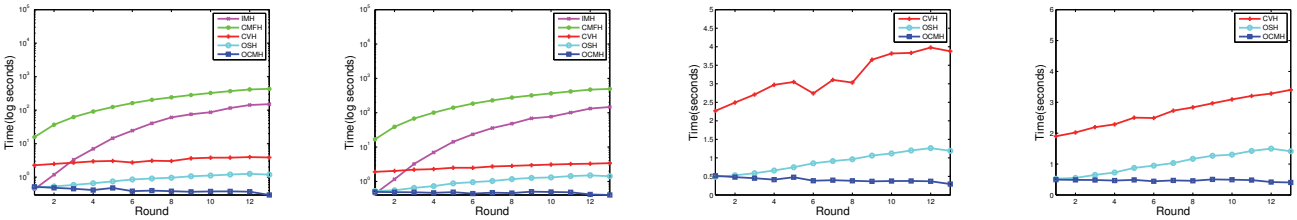
In our experiments, two types of cross-modal retrieval tasks: image query and text query, are considered for evaluation. The whole online retrieval process contains several rounds. At each round, a new data chunk is added to the database, and we evaluate the retrieval performance. According to the number of data chunks, MIR Flickr has 13 rounds in total, and NUS-WIDE has 27 rounds.

CVH is batch learning based hashing, at each round, we use all database data to retrain hash functions and codes. CMFH and IMH are also batch learning based methods, but they need tremendous time and memory to use whole database for learning. Thus in training process of IMH, we randomly select 10% database data on both datasets. CMFH is more efficient than IMH, thus for the training of



(a) Image query (32 bits) (b) Text query (32 bits) (c) Image query (64 bits) (d) Text query (64 bits)

Figure 4: The MAP scores of MIR Flickr at each round, with 32 and 64 bits of hash codes.



(a) 32 bits (b) 64 bits (c) 32 bits (d) 64 bits

Figure 5: Time comparison on MIR Flickr with 32 and 64 bits, both log seconds and seconds are used.

CMFH, we select 10% database data on NUS-WIDE, and all database data on MIR Flcikr. OSH can learn hash functions online, but when it computes hash codes, all database data should be used. In addition, OSH is not designed for multi-modal data, thus we have to slightly improve OSH for cross-modal hashing. In the hashing process of OSH, we concatenate image feature x_1 and text feature x_2 to form a single feature x, then the learned hashing weight matrix $W = [W_1^T, W_2^T]^T$.

Fig. 2 shows the MAP scores of all compared methods on NUS-WIDE at each round, with 32 and 64 bits. We can find that the MAP scores of OCMH are consistently increased with the increase of rounds, which illustrates that cross-modal correlation can be improved by OCMH at each round. Generally, online learning may lose some precision to achieve efficiency. But Fig. 2 shows an interesting phenomenon, that the online learning based OCMH performs even better than batch learning based CVH, IMH and CMFH. The reason is that OCMH can analyze latent cross-modal correlation by SLC, which is ignored by other CMH methods. As a result, OCMH can learn more cross-modal correlation while guarantee the learning efficiency. OCMH also outperforms OSH, the reason is that cross-modal correlation cannot be learned by OSH which is not designed for multi-modal data.

Fig. 3 shows the learning time of all compared methods on NUS-WIDE with 32 and 64 bits. Since the learning time of IMH and CMFH is much larger than other methods. The left two figures use log seconds which are the log value of seconds, to demonstrate the time comparison of all methods. From these two figures we can find OCMH costs the least learning time, and IMH and CMFH are inefficient in the online scenario. The right two figures only show the seconds of OCMH, CVH and OSH, we can find from them that the time cost of OCMH at each round is not increased. This result confirms that the time complexity of OCMH is independent with the database size which is increased at each round. In addition, we can find the time cost of OSH is increased linearly at each round. OSH can learn hash functions online, but it has to compute all database hash codes at each round, thus it is less efficient for the online retrieval.

Fig. 4 and Fig. 5 show the MAP scores and learning time of all compared methods on MIR Flickr at each round, with 32 and 64 bits. From them, we can obtain similar results to NUS-WIDE. OCMH gains the best MAP scores at most rounds, and it costs the least learning times. OCMH cannot significantly outperform CVH in terms of MAP score, but it costs much less learning time than CVH. Therefore OCMH can better solve online web image retrieval. In addition, OCMH achieves more significant performance on NUS-WIDE, which implies that OCMH is more suited to larger database.

Conclusion

In this paper we propose OCMH for the effective and efficient retrieval of multi-modal web images. OCMH is superior to traditional CMH in that it can effectively learn the streaming web data online. Besides, it specially considers efficient updating of hash codes which is not solved by previous online hashing methods. OCMH solves the updating of hash codes by transforming it to the efficient updating of SLC and transfer matrix. Moreover, SLC encodes the latent cross-modal correlation which can improve the effect of cross-modal analysis. Then, an efficient online optimization algorithm, whose time complexity is independent with database size, is proposed for updating SLC and hash

functions. Experimental results on two web image datasets demonstrate both the effectiveness and efficiency of OCMH.

Acknowledgement

Jialie Shen and Lei Zhu are supported by Singapore MOE tier 2 one.

References

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of ACM International Conference on Image and Video Retrieval (CIVR 09)*, 48. ACM.

Costa Pereira, J.; Coviello, E.; Doyle, G.; Rasiwasia, N.; Lanckriet, G. R.; Levy, R.; and Vasconcelos, N. 2014. On the role of correlation and abstraction in cross-modal multimedia retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):521–535.

Ding, G.; Guo, Y.; and Zhou, J. 2014. Collective matrix factorization hashing for multimodal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, 2083–2090. IEEE.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multimodal semi-supervised learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 10)*, 902–909. IEEE.

Huang, L.-K.; Yang, Q.; and Zheng, W.-S. 2013. Online hashing. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 13)*, 1422–1428. AAAI Press.

Jain, P.; Kulis, B.; Dhillon, I. S.; and Grauman, K. 2009. Online metric learning and fast similarity search. In *Advances in Neural Information Processing Systems (NIPS 09)*, 761–768.

Kumar, S., and Udupa, R. 2011. Learning hash functions for cross-view similarity search. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 11)*, volume 22, 1360.

Leng, C.; Wu, J.; Cheng, J.; Bai, X.; and Lu, H. 2015. Online sketching hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 15)*, 2503–2511.

Liberty, E. 2013. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 13)*, 581–588. ACM.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1997. Kernel principal component analysis. In *Proceedings of Artificial Neural Networks (ICANN 97)*. Springer. 583–588.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 08)*, 650–658. ACM.

Song, J.; Yang, Y.; Yang, Y.; Huang, Z.; and Shen, H. T. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD 13)*, 785–796. ACM.

Wang, J.; Shen, H. T.; Song, J.; and Ji, J. 2014. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*.

Wang, J.; Kumar, S.; and Chang, S.-F. 2012. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(12):2393–2406.

Weiss, Y.; Torralba, A.; and Fergus, R. 2009. Spectral hashing. In *Advances in Neural Information Processing Systems (NIPS 09)*, 1753–1760.

Wu, B.; Yang, Q.; Zheng, W.-S.; Wang, Y.; and Wang, J. 2015. Quantized correlation hashing for fast cross-modal search. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 15)*, 3946–3952. AAAI Press.

Xie, L.; Zhu, L.; Pan, P.; and Lu, Y. 2015. Cross-modal self-taught hashing for large-scale image retrieval. *Signal Processing*.

Xie, L.; Pan, P.; and Lu, Y. 2015. Analyzing semantic correlation for cross-modal retrieval. *Multimedia Systems* 21(6):525–539.

Zhai, X.; Peng, Y.; and Xiao, J. 2013. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI 13)*.

Zhang, D., and Li, W.-J. 2014. Large-scale supervised multimodal hashing with semantic correlation maximization. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 14)*, 2177–2183.

Zhang, D.; Wang, J.; Cai, D.; and Lu, J. 2010. Self-taught hashing for fast similarity search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 10)*, 18–25. ACM.

Zhou, J.; Ding, G.; and Guo, Y. 2014. Latent semantic sparse hashing for cross-modal similarity search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR 14)*, 415–424. ACM.

Zhu, X.; Huang, Z.; Shen, H. T.; and Zhao, X. 2013. Linear cross-modal hashing for efficient multimedia search. In *Proceedings of the 21st ACM International Conference on Multimedia (ACM MM 13)*, 143–152. ACM.

Zhu, L.; Shen, J.; Jin, H.; Zheng, R.; and Xie, L. 2015. Content-based visual landmark search via multi-modal hypergraph learning. *IEEE Transactions on Cybernetics*.

Zhu, L.; Shen, J.; and Xie, L. 2015. Topic hypergraph hashing for mobile image retrieval. In *Proceedings of the 23rd ACM Conference on Multimedia (MM 15)*, 843–846. ACM.