Nonlinear Discrete Cross-Modal Hashing for Visual-Textual Data

Dekui Ma
Dalian University of Technology

Jian Liang and Ran He
Chinese Academy of Sciences Institute of Automation

Xiangwei Kong
Dalian University of Technology

Discrete cross-modal hashing is a supervised method that exploits classification tasks to learn heterogeneous binary codes. DCMH also updates the binary codes for each modality and learns discrete hashing codes bit by bit, making it promising for large-scale datasets.

Hashing is an effective technique for approximate nearest-neighbor search. Because hashing methods have low storage costs, they've drawn considerable attention in the big data era, with numerous methods being proposed in the past few years.[1,2] Traditional hashing methods focus on homogenous data forms. However, the ever-increasing amount of multimedia data on social websites and mobile applications are naturally surrounded by textual information, including descriptions, tags, and user comments. To capture these heterogeneous image and text modalities, researchers have proposed numerous cross-modal retrieval methods.[3–5] Furthermore, the binary codes for cross-modal retrieval—that is, cross-modal hashing—have been exploited to meet the needs of storage usage and training time.[6–8]

Most cross-modal hashing methods focus on how to design hashing functions to preserve data similarities in the Hamming space (see the "Related Work in Cross-Modal Hashing" sidebar for more information). However, these approaches typically relax the binary constraints to simplify the optimization process, thereby degrading retrieval performance.

Inspired by unimodal hashing methods,[2] we developed a discrete hashing method for cross-modal retrieval called *discrete cross-modal hashing*. DCMH employs an iterative optimization method to learn hashing functions without relaxing the discrete constraints. We formulate the objective function by reconstructing the semantic intersimilarity matrix and regard the learned binary codes as ideal features for intramodal classification. To simplify the optimization process, DCMH uses linear regression to form both hashing functions and the classification matrix. To address the NP-hard binary optimization problem, we apply the *discrete cyclic coordinate descent* method.[2] The overall objective function consists primarily of two intramodal hashing functions and one intersimilarity reconstruction term; the intramodal hashing function primarily relies on binary features classification-error criterion.

Here, to show the effectiveness of our hashing model and optimization methods, we describe the traditional relax-and-threshold solution (dubbed *DCMH_rat*) and compare it with DCMH (see the sidebar for more on the relax-and-threshold solution).

This article expands on our previous conference paper[9] as follows. First, we provide a relaxation solution with our objective function and compare it with the formerly proposed discrete solution to further verify the advantages of the proposed objective function and the benefits brought by discrete optimization. Second, to further show the effectiveness of our proposed methods, we add two novel large-scale datasets, including a multilabel dataset, to the experiments. Finally, we evaluate the intramodal retrieval performances—that is, image-to-image and text-to-text—to prove our cross-modal model's generalization abilities.

Discrete Cross-Modal Hashing

In this section, we explain the proposed method and describe the associated optimization algorithm.

Problem Definition

For simplicity, we assume here that there are only two modalities, but DCMH can be easily

1070-986X/17/$33.00 © 2017 IEEE

Related Work in Cross-Modal Hashing

Existing cross-modal hashing methods can be categorized as unsupervised and supervised methods. One classical unsupervised method extended spectral hashing to the multimodal setting by minimizing the weighted distance.[1] Guiguang Ding and his colleagues used collective matrix factorization for different modalities to obtain the hashing functions with latent a factor model.[2]

Supervised methods usually achieve much better performance because they use semantic labels or pairwise relationships to learn the discriminative hashing functions via label-similarity preserving criterion.

Jingkuan Song and his colleagues considered the differences between each modality by exploring single modality correlations and keeping the different modalities' codes consistent.[3] Other researchers proposed maximizing the semantic correlation and further optimizing the objective function in a greedy way for large-scale datasets.[4] In addition, Yueting Zhuang and his colleagues used neural network models for cross-media hashing,[5] while Xiaobo Shen and his colleagues exploited matrix factorization for multiview data.[6] Recently, Dekui Ma and his colleagues proposed a simple two-step approach and obtained impressive retrieval performances on various benchmark datasets, where the binary codes obtained via unimodal hashing methods were considered as unified codes for both modalities.[7]

In addition to data similarity preservation, quantization qualities are also crucial for hashing-based retrieval methods, as proven in the classical unimodal hashing papers.[8] Similar to the unimodal hashing methods, cross-modal hashing approaches have inevitable binary constraints, which make the objective function challenging to optimize. To make the optimization problem feasible, most hashing approaches adopt a two-step *relax-and-threshold* strategy: first, they learn real hashing functions to relax the constraints, and then they threshold them to obtain the discrete codes. However, this trick brings nonnegligible quantization errors, and is thus suboptimal.

Recently, many research efforts—including the classic iterative quantization (ITQ) method—have aimed to minimize quantization.[9] By introducing a rotation matrix, ITQ minimized quantization errors and thus obtained better hashing projection matrices. By introducing an auxiliary variable for discrete codes, supervised discrete hashing (SDH)[10] reformulated the objective function and obtained an efficient discrete solution via cyclic coordinate descent. Work by Go Irie and her colleagues was pioneering in its focus on quantization errors for cross-modal hashing.[11] Their efforts sought binary quantizers for each modality by simultaneously minimizing the binary quantization problem and subspace learning.

References

1. S. Kumar and U. Raghavendra, "Learning Hash Functions for Cross-View Similarity Search," *Proc. 20th Int'l Joint Conf. Artificial Intelligence* (IJCAI), 2011, pp. 1360–1365.
2. G. Ding, Y. Guo, and J. Zhou, "Collective Matrix Factorization Hashing for Multimodal Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 2083–2090.
3. J. Song et al., "Inter-Media Hashing for Large-Scale Retrieval from Heterogeneous Data Sources," *Proc. 2013 ACM SIGMOD Int'l Conf. Management of Data*, 2013, pp. 785–796.
4. D. Zhang and W.J. Li, "Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization," *Proc. 28th AAAI Conf. Artificial Intelligence* (AAAI), 2014, pp. 2177–2183.
5. Y. Zhuang et al., "Cross-Media Hashing with Neural Networks," *Proc. 22nd ACM Int'l Conf. Multimedia*, 2014, pp. 901–904.
6. X. Shen et al., "Multi-View Latent Hashing for Efficient Multimedia Search," *Proc. 23rd ACM Int'l Conf. Multimedia*, 2015, pp. 831–834.
7. D. Ma et al., "Frustratingly Easy Cross-Modal Hashing," *Proc. 2016 ACM on Multimedia Conference*, 2016, pp. 237–241.
8. Y. Gong et al., "Angular Quantization-Based Binary Codes for Fast Similarity Search," *Advances in Neural Information Processing Systems*, 2012, pp. 1196–1204.
9. Y. Gong et al., "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, 2013, pp. 2916–2929.
10. F. Shen et al., "Supervised Discrete Hashing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 37–45.
11. G. Irie, H. Arai, and Y. Taniguchi, "Alternating Co-Quantization for Cross-Modal Hashing," *Proc. IEEE Int'l Conf. Computer Vision*, 2015, pp. 1886–1894.

extended to more. Assume $X = \{x_i\}_{i=1}^{n}, x_i = \{x_i^1, x_i^2\}$ represents n data points of two different modalities, where $x_i^1 \in R^m$ is an m-dimensional image feature, and $x_i^2 \in R^d$ is a d-dimensional text feature vector. Given the code length k, our goal is to learn hashing functions $\mathrm{f}_q(\cdot)$ that map the original continuous features x_i^q to binary codes $h_i^q \in \{-1, 1\}^k, q = \{1, 2\}$. Here, for each modality, we adopt the simple linear hashing function $f_q(\cdot) = sgn(W_q^T x)$, where matrices W_q are the projection matrices that we need to learn. $Y \in \{0,1\}^{c \times n}$ denotes the label matrix and $y_i \in R^c$ denotes the ith label vector, where c is the number of semantic categories in the dataset.

Intermodality Similarity Preservation

Unlike previous unified binary-code-based methods,[10,11] we used two binary matrices, H_1 and H_2, each of which represents a separate binary space and connects it with intermodality similarity-preserving terms. This should let heterogeneous points from different modalities in the projected binary space be close to each other. The heterogeneous similarity affinity matrix S is directly generated from Y, while $s_{i,j} = 1$ indicates that the ith and jth objects share at least one common semantic label; otherwise $s_{i,j} = -1$. For multilabel datasets, we can use a more complex similarity metric, such as cosine distance; however, we found that cosine distance does not improve performance.

We define the basic object function on intermodal similarity preservation as follows:

$$\min ||H_1^T H_2 - cS||_F^2, \tag{1}$$

where H_1, $H_2 \in \{-1, +1\}^{k \times n}$ are the learned hashing codes, and each binary code $h_i^q = sgn(P_i^T x_i^q)$ and $P_1 \in \mathrm{R}^{m \times k}$, $P_2 \in \mathrm{R}^{d \times k}$ are learned hashing projection matrices. The inner products of H_1 and H_2 reflect the opposite of their Hamming distance (to some extent). We adopt the square loss for similarity reconstruction, which is widely used in hashing methods.

Intramodality Similarity Preservation

In addition to similarity preservation across modalities, we aim to preserve the similarity within each modality, which is also the main focus of unimodal hashing methods.

To simplify the optimization problem, we adopt an approach similar to that of Fumin Shen and his colleagues[2] to obtain hashing functions—that is, we use simple linear regressions. To leverage the semantic labels to hashing function learning, we optimize the learned binary codes into a classification task. Our goal is that the learned hashing codes be well classified along with the semantic labels.

When we consider only one modality, the objective function of classification with hidden binary codes can be written as

$$\min_{W,H} \sum_{i=1}^{n} L(y_i, W^T h_i) + \lambda ||W||_F^2, \tag{2}$$

where each code $h_i = sgn(P^T x_i)$, $L(\cdot)$ is the loss function of a classification model, and λ is the regularization parameter. Because we can select any appropriate loss function for L, we chose l_2 loss due to its simplicity. By introducing the matrix expression, we can rewrite the problem in Equation 2 as

$$\begin{aligned} \min{}_{W,P,H \in \{\pm 1\}^{k \times n}} & ||Y - W^T H||_F^2 \\ & + \eta ||H - P^T X||_F^2 + \lambda R(W, P). \end{aligned} \tag{3}$$

To avoid trivial solutions, we use $R(\cdot) = \| \cdot \|_F^2$ as regularization terms.

Overall Formulation and Optimization

Combining the inter- and intramodality similarity preservation terms in Equations 1 and 3, we get the final objective function of the proposed DCMH:

$$\begin{aligned} \min{}_{H,W,P}\, G = & \sum_{i=1,2} ||Y - W_i^T H_i||_F^2 + \eta ||H_i - P_i^T X_i||_F^2 \\ & + \lambda R(W_i, P_i) + \gamma ||H_1^T H_2 - cS||_F^2 \\ & \text{s.t.} \quad H_i \in \{-1, +1\}^{k \times n}. \end{aligned} \tag{4}$$

Here, η, λ, and γ are tradeoff parameters.

Nonlinear embedding beforehand can boost the performances of linear methods; it is also scalable for high-dimensional data matrices. Hence, we adopt a simple yet effective nonlinear technique[1] as follows: $F(x) = sgn(P^T \phi(x))$, where $\phi(x) = [\exp(\|x - z_1\|^2/\sigma), \cdots, \exp(\|x - z_l\|^2/\sigma)]$. Here, $\{z_j\}_{j=1}^{l}$ are the randomly selected l landmark points and σ is the kernel width.

Obviously, the objective function in Equation 4 is nonconvex. Fortunately, the subproblem for any of the six variables is convex while fixing the other five variables. Thus, we can obtain local optima in an alternating optimization manner. DCMH alternately updates the six variables by following the listed three steps until convergence. We found that only a few iterations within each modality can give reasonably stable performances.

P-Step. When we fix H and W and let $\frac{\partial G}{\partial P_i} = 0$, we obtain

$$P_i = \left(\phi(X_i) \phi(X_i)^T + \lambda I \right)^{-1} \phi(X_i) H_i^T, \tag{5}$$

where I is an identity matrix. This step can be seen as a simple least-square linear regression.

W-Step. When we fix H and P and let $\frac{\partial G}{\partial W_i} = 0$, we obtain

$$W_i = (H_i H_i^T + \lambda I)^{-1} H_i Y^T. \tag{6}$$

Here, we can also obtain a closed-form solution for each W_i.

H-Step. When we fix W and P, we can rewrite Equation 4 as

$$\min_{H\in\{\pm 1\}^{k\times n}} ||Y - W_i^T H_i||_F^2 + \eta ||H_i - P_i^T \phi(X_i)||_F^2 + \gamma ||H_1^T H_2 - cS||_F^2. \quad (7)$$

Given the discrete constraints, solving H becomes an NP-hard problem. Most existing methods directly relax this constraint and binarize the optimal continuous solution, while other methods try to optimize it by introducing a sigmoid function. However, we attempt to learn the binary codes along with the discrete constraints. One naive approach is enumeration, but it is uncomputable. Here, for H_1, Equation 7 is directly decomposed as follows:

$$\min_{H_1\in\{\pm 1\}^{k\times n}} ||W_1 H_1||_F^2 - 2Tr(H_1^T(W_1 Y + \eta P_1^T \phi(X_1)) + \gamma c H_2 S)). \quad (8)$$

We then adopt the *discrete cyclic coordinate* (DCC) descent method[2] to solve this discrete optimization problem—that is, the *i*th column of H is updated when the remains are fixed. We adopt the DCC descent method to optimize Equation 8 in several iterations.

Optimization with the Relax-and-Threshold Strategy

To measure the respective contributions of the proposed hashing model and optimization method shown in Algorithm 1 (Figure 1), we also give an optimization algorithm for DCMH with the relax-and-threshold strategy, DCMH_rat.

Generally, DCMH_rat also adopts an iterative updating process, which instantly optimizes a relaxed continuous objective function by dropping the discrete constraints. Compared with Algorithm 1, only H-Step needs to be adjusted. As we show in the following experiments, a specific gap exists between DCMH and DCMH_rat, where DCMH_rat can suffer from larger quantization errors.

H-Step. When we fix W and P, we can rewrite Equation 4 as

$$G(H_i) = ||Y - W_i^T H_i||_F^2 + \eta ||H_i - P_i^T \phi(X_i)||_F^2 + \gamma ||H_i^T H_j - cS||_F^2. \quad (9)$$

To obtain optimal H_i, H_i is updated by $H_i + d$ in each iteration, then the corresponding problem is defined as *arg min* $G(H_i + d)$. Taylor expansion is further applied to approximate $G(H_i + d)$ as

$$G(H_i + d) \approx G(H_i) + G'(H_i)d + 1/2G''(H_i)d^2, \quad (10)$$

where $G'(H_i)$ and $G''(H_i)$ are the first- and second-order derivatives of G about H_i, and their detailed expressions are listed as

$$\begin{aligned} G'(H_i) &= (W_i W_i^T + \eta I + \gamma H_j H_j^T)H_i \\ &\quad - (W_i Y + \eta P_i^T \phi(X_i) + \gamma c H_j S) \quad (11) \\ G''(H_i) &= W_i W_i^T + \eta I + \gamma H_j H_j^T. \end{aligned}$$

Finally, given the classical Newton method, H_i is updated each time as follows:

$$H_i(t+1) = H_i(t) + \alpha d. \quad (12)$$

Here, the direction vector $d = G'(H_i)/G''(H_i)$, and α is the step-size parameter, controlling the convergence of the iterative updating process.

Time Complexity and Convergence Analysis

Because DCMH adopts an iterative optimization, P-Step and W-Step are classical linear regression solutions that occupy $O(nl^2k)$ and $O(nd^2k)$.
H-Step occupies $O(tk^2n + tk^2c)$ for each iteration, where t is the number of iterations. The overall computational complexity is $O(T(nk^2))$, where T is the number of external iterations. In the testing phase, the complexity of generating hashing codes is constant, with $O(mk)$ for an image query and $O(dk)$ for a text query. Hence, DCMH has a linear complexity to dataset size n and is flexible for large-scale datasets.

To seek an optimal solution, the variables P, W, and B are alternately learned for several iterations. The objective function in Equation 4 is minimized in each step; we show the convergence analysis of DCMH as

$$G(P^{(t)}, W^{(t)}, B^{(t)}) \geq G(P^{(t+1)}, W^{(t)}, B^{(t)}) \geq G(P^{(t+1)}, W^{(t+1)}, B^{(t)}) \geq G(P^{(t+1)}, W^{(t+1)}, B^{(t+1)}), \quad (13)$$

where $P^{(t)}$, $W^{(t)}$, and $B^{(t)}$ are matrices in the *t*th iteration.

Algorithm 1 shows the proposed DCMH procedure.

Experiments

We compare our DCMH with baseline methods on five benchmark datasets, with visual features for images and textual features for user tags or webpages.

Datasets and Setting

The Wiki dataset (www.svcl.ucsd.edu/projects/crossmodal) consists of 2,866 text-image

```
Input: Data matrices $X^{(t)}$, $t = 1, 2$, semantic label matrix $Y$ and hash code length $k$.
Output: Hash projection matrices $P_i$, $i = 1, 2$.
Procedure:
1. Randomly select $l$ objects to get the nonlinear embedding data $\phi(X)$ via the
   RBF kernel function;
2. Initialize $H$ as $\{-1, 1\}^{k \times n}$ randomly;
3. Repeat:
        a) Obtain $P_1$ and $P_2$ via Equation 5;
        b) Obtain $W_1$ and $W_2$ via Equation 6;
        c) Iteratively solve $H_1$ and $H_2$ via Equation 8 with the help of DCC;
   Until reaching convergence or maximum iterations.
```

Figure 1. Algorithm 1: The pseudo code of discrete cross-modal hashing (DCMH).

Table 1. Dataset characteristics.

Dataset	Training/testing	Image/text	Class	Labels
Wiki	2,173/693	128/10	10	Single
Wiki+	2,173/693	4,096/5,000	10	Single
LabelMe	2,014/672	512/470	8	Single
VOC+	2,808/2,841	4,096/399	20	Single
MIRFLickr	15,902/836	150/500	24	Multilabel
INRIA-Websearch	10,332/4,366	4,096/1,000	100	Single

documents labeled as one of 10 semantic categories. Wiki+[4] shares the same settings as the Wiki dataset, but its images are 4,096-dimensional convolutional neural network (CNN) features, and its texts are 5,000-dimensional bag of words (BoW) features on the term frequency-inverse document frequency (TF-IDF) weighting scheme.

The LabelMe outdoor dataset consists of 2,686 fully annotated outdoor images from eight scene categories. Following earlier work,[11] we randomly split the dataset into training/testing sets using a 3:1 ratio.

The PASCAL Visual Object Classes (VOC)+ dataset includes 2,808 training and 2,841 testing data; the images are associated with only a single label.[12] Following earlier work,[10] we use the CNN features instead of original gist features for images.

The MIRFLickr dataset is composed of 16,738 instances collected from the social photography website Flickr. Following earlier work,[11] we randomly split the dataset into a training set and a testing set, with 15,902 and 836 (5 percent), respectively. This dataset includes 24 ground-truth labels (tags), and each instance might be associated with multiple labels.

The INRIA-Websearch dataset (http://lear.inrialpes.fr/~krapac/webqueries/webqueries.html) contains 71,478 pairs of web images and text annotations from 353 categories, including actors, logos, and landmarks. We obtained 14,698 pairs as in earlier work[13] and randomly split them into training/testing sets (3:1).

Table 1 shows basic information for each dataset.

Experiment Setting

Here, we introduce some related cross-modal hashing methods and compare them with our DCMH and DCMH_rat on some common evaluation schemes, such as the mean average precision (MAP) and normalized discounted cumulative gain (NDCG).

Baseline methods. We compare DCMH with several cross-modal hashing methods, including unsupervised methods, such as cross-view hashing (CVH)[3] and collective matrix factorization hashing (CMFH),[6] and supervised ones, such as intermedia hashing (IMH)[14] and sequential semantic correlation maximization (SCM_Seq).[8] All source codes are available publicly, and all parameters are set to be consistent with their original presentation. We consider IMH as supervised by training all instances. For DCMH and DCMH_rat, l is fixed at 500. All results are averaged over four runs to eliminate the influence of random initialization, and we use post hoc tests to compare our method with the other methods. We ran all experiments on a workstation with a 2.60 GHz Intel Xeon E5-2650 CPU and 32.0 Gbytes RAM.

Evaluation scheme. Some previous works use the training set as a gallery for cross-modal learning, but the high-retrieval performance might be overfitting. Given this and following other efforts,[4,5] we adopt the testing set as a gallery here.

We adopt MAP, which is widely used for retrieval tasks, to measure the performance of all methods. The top r average precision (AP) can be defined as

$$AP@r = \frac{1}{L}\sum_{i=1}^{r} P(i) \times \delta(i), \qquad (14)$$

where L is the number of relevant instances, $P(i)$ denotes the precision value, and $\delta(i)$ is an

Table 2. Mean average precision (%) for the top 50 retrieved instances for image and text queries on Wiki and LabelMe. (Results in bold represent the best performance.)

		Wiki				LabelMe			
Number of bits		16	24	32	64	16	24	32	64
Image query	CVH	27.05	26.04	26.02	24.68	36.92	36.58	35.32	35.46
	CMFH	32.47	34.01	34.81	35.85	40.09	46.71	60.20	50.12
	IMH	23.99	23.55	23.33	21.43	46.14	43.01	40.41	35.57
	SCM_Seq	34.28	35.24	34.57	36.23	67.10	68.56	70.48	72.53
	DCMH_rat	**37.24**	37.89	**41.49**	38.50	73.17	76.59	78.43	79.63
	DCMH	36.81	**38.71**	41.05	**43.44**	**76.00**	**78.36**	**78.88**	**79.66**
Text query	CVH	23.13	23.21	22.01	20.12	38.99	39.12	38.35	37.58
	CMFH	30.63	32.96	33.98	32.67	40.87	47.75	48.65	49.54
	IMH	24.36	22.91	21.62	20.40	48.64	44.81	42.09	35.90
	SCM_Seq	31.37	32.24	32.41	33.67	74.56	75.11	76.79	80.28
	DCMH_rat	30.16	**34.31**	**36.22**	33.61	80.61	84.73	84.64	85.67
	DCMH	**37.88**	34.24	33.51	**36.72**	**85.57**	**87.31**	**86.10**	**88.03**

indicator function; r is the number of retrieved instances and is fixed at 50 here. We consider a retrieved instance as a true neighbor if it shares at least one common semantic label with the query.[8,11]

In addition, NDCG is a standard and commonly used metric for ranking-based datasets. The NDCG value for the top k results is defined as

$$NDCG@k = \frac{1}{Z}\sum_{i=1}^{k}\frac{2^{r_i}-1}{log_2(i+1)}, \qquad (15)$$

where r_i is a relevance index between the query and the ith ranked sample, and Z is a normalization term that ensures the optimal ranking with an NDCG score of 1. For multilabel datasets, the number of shared labels is seen as the relevance value here.

Experimental Results and Discussion

All of the datasets in Table 1 are summarized into three categories:

- *small-scale* datasets (Wiki and LabelMe),
- *high-dimensional* datasets (Wiki+ and VOC+), and
- *large-scale* datasets (MIRFLickr and INRIA-Websearch).

When implementing DCMH for large-scale datasets, we randomly select 5,000 instances as training sets.

Wiki and LabelMe results. Table 2 shows the MAP values on the small-scale datasets, with hashing bits in the range of {16, 24, 32, 64}. DCMH and DCMH_rat significantly outperform other methods in these two datasets for both text and image queries. Compared with the second best method, SCM_Seq, the maximum gains of DCMH reach 19.9 percent for image query and 20.7 percent for text query on Wiki, and, on average, more than 12 percent for image query and 13 percent for text query on LabelMe. DCMH also performs better with longer codes because more information can be encoded, and it almost always beats DCMH_rat, except for text query at 32 bits on the Wiki dataset. On these two datasets, DCMH outperforms other baseline methods, at a significance level of 95 percent.

Wiki+ and VOC+ results. Due to their dramatic performance, high-dimensional features—especially CNN full-connected features—have been increasingly popular. To better exploit DCMH's performance, we also report the results for high-dimensional datasets (see Table 3). The proposed DCMH and DCMH_rat again outperform other baseline methods.

For VOC+, DCMH obtains nearly 100 percent MAP value at 64 bits, and the gain obtained by DCMH is significant over both retrieval tasks on Wiki+. Although DCMH_rat does not perform quite as well as DCMH, it achieves comparable performance and

Table 3. Mean average precision (%) for the top 50 retrieved instances for image and text queries on Wiki+ and VOC+. (Results in bold represent the best performance.)

		Wiki+				VOC+			
Number of bits		16	24	32	64	16	24	32	64
Image query	CVH	16.97	16.97	16.96	16.96	50.91	52.74	55.45	53.69
	CMFH	29.71	31.11	31.55	32.17	22.84	23.58	23.44	24.06
	IMH	33.19	33.13	32.49	30.88	64.03	62.99	61.29	58.70
	SCM_Seq	42.26	46.66	46.66	48.59	83.68	88.91	90.42	91.74
	DCMH_rat	**52.31**	57.55	57.58	58.65	**95.52**	**97.53**	96.93	98.37
	DCMH	53.39	**58.43**	**60.52**	**61.16**	90.89	97.13	**98.94**	**99.11**
Text query	CVH	18.14	16.80	16.37	18.00	19.33	19.09	17.01	16.54
	CMFH	29.60	30.94	31.12	32.12	22.01	20.95	24.70	23.72
	IMH	33.40	33.87	32.99	31.37	54.95	49.89	43.79	34.98
	SCM_Seq	45.75	48.67	47.86	51.95	74.48	76.55	75.61	75.36
	DCMH_rat	53.33	54.90	56.08	59.16	**89.78**	**94.70**	94.08	93.27
	DCMH	**55.48**	**58.01**	**60.27**	**61.25**	87.58	93.24	**95.83**	**96.43**

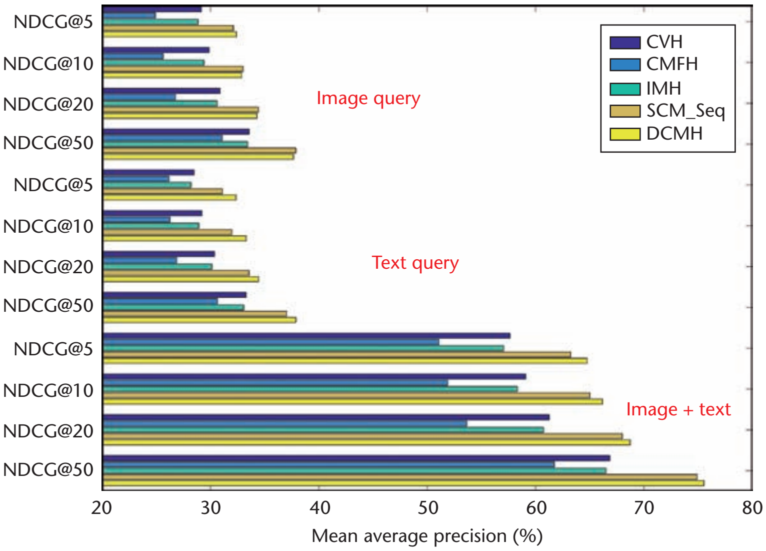

Figure 2. Normalized discounted cumulative gain (NDCG) results on the MIRFLickr datasets for the image and task query tasks. The top five are results for image query, the middle five are results for text query, and the last five are the summation of these two queries.

outperforms other methods. Compared with the Wiki results, all methods significantly improve on Wiki+, which can be attributed mostly to the advantages of CNN features. DCMH's maximum gains reach 50.9 percent for images query and 79.8 percent for text query at 32 bits, while SCM_Seq reaches 34.9 and 54.3 percent, respectively. DCMH also outperforms other baseline methods, at a significance level of 95 percent.

INRIA-Websearch and MIRFLickr results. Table 4 shows results on the INRIA-Websearch and MIRFLickr large-scale datasets. As expected, DCMH outperforms DCMH_rat, except at 64 bits. SCM_Seq obtains the best retrieval performance for image query on MIRFLickr. However, DCMH is still competitive with SCM_Seq and, for other tasks, DCMH has the best performance. Moreover, all methods perform well on the MIRFLickr dataset, which can be attributed to its multilabel property.

To measure the performances on the multilabel MIRFLickr dataset more accurately, Figure 2 shows the NDCG scores. DCMH achieves the best scores on the text-query task and obtains competitive scores on the image-query task. SCM_Seq adopts the cosine similarity, which might explain why it returns more "close text" tags. For the text-query task, the traditional feature representations of images cannot match the full tag information. However, DCMH consistently outperforms SCM_Seq when considering both query tasks.

Nonlinear embedding results. To evaluate the effectiveness of the proposed nonlinear DCMH, we adopt the nonlinear trick described earlier to boost performance for linear baseline methods. Here, CMFH and SCM_Seq achieve lower MAP values, while other methods (CVH and IMH) achieve better performance. For the MIRFLickr dataset, SCM_Seq again achieves the best performance. Due to the multilabel property, the NDCG values are more reliable than

Table 4. Mean average precision (%) for the top 50 retrieved instances for image and text queries on INRIA-Websearch and MIRFLickr. (Results in bold represent the best performance.)

		INRIA-Websearch				MIRFLickr			
Number of bits		16	24	32	64	16	24	32	64
Image query	CVH	28.40	31.96	35.53	40.87	63.74	63.23	62.88	61.79
	CMFH	33.45	39.13	40.63	47.32	57.02	57.13	56.65	56.45
	IMH	29.40	30.48	35.37	42.22	63.38	62.68	63.63	61.71
	SCM_Seq	38.12	38.48	35.03	40.50	**69.19**	**69.49**	**70.02**	70.37
	DCMH_rat	46.76	50.04	52.05	**59.14**	65.88	66.71	68.22	67.43
	DCMH	**49.48**	**52.76**	**55.48**	51.00	68.68	68.99	69.96	**71.32**
Text query	CVH	29.06	34.98	39.65	46.44	63.48	63.32	62.83	61.13
	CMFH	33.52	40.16	44.51	54.37	56.91	57.24	57.11	57.07
	IMH	30.50	33.31	39.92	50.08	63.76	62.73	63.03	61.69
	SCM_Seq	29.03	33.11	38.05	46.11	68.57	69.25	69.55	69.85
	DCMH_rat	48.13	54.39	58.64	**65.58**	66.51	67.84	68.89	**70.09**
	DCMH	**52.49**	**56.64**	**61.13**	53.06	**68.59**	**69.97**	**69.59**	69.95

Table 5. Mean average precision (%) for the top 50 retrieved instances for image and text queries with nonlinear embedding for LabelMe and Wiki+. (Results in bold represent the best performance.)*

		LabelMe				Wiki+			
Number of bits		16	24	32	64	16	24	32	64
Image query	CVH	51.17	49.02	48.30	44.86	17.53	24.50	25.06	25.19
	CMFH	26.49	26.39	26.50	26.50	17.53	17.36	17.35	17.50
	IMH	46.41	41.28	38.73	36.30	34.14	34.43	34.82	31.14
	SCM_Seq	53.49	56.88	56.13	54.67	31.36	35.34	31.69	22.42
	DCMH	**76.00**	**78.36**	**78.88**	**79.66**	**53.39**	**58.43**	**60.52**	**61.16**
Text query	CVH	52.59	50.02	49.60	45.81	24.39	25.07	25.44	25.89
	CMFH	25.96	25.92	25.88	26.01	18.26	18.36	18.58	18.61
	IMH	48.64	44.81	42.09	35.90	32.67	35.38	35.00	32.19
	SCM_Seq	41.28	56.35	50.56	48.71	19.32	26.66	21.78	17.65
	DCMH	**85.57**	**87.31**	**86.10**	**88.03**	**55.48**	**58.01**	**60.27**	**61.25**

*All baseline methods adopt the same nonlinear embedding trick.

MAP values, where DCMH outperforms SCM_Seq consistently (see Figure 2). Moreover, our DCMH performs better than other methods on the smallest code bits. Once again, as Tables 5 and 6 show, our DCMH achieves much better accuracies than the baseline methods.

Results for intramodal retrieval. As Table 7 shows, in addition to cross-modal retrieval, we also compare different methods at 32 bits in the intramodal retrieval tasks—that is, image-to-image (I2I) and text-to-text (T2T). For both tasks, DCMH significantly outperforms the other two supervised methods except for on the T2T task using the VOC+ dataset. Because VOC+'s textual features are so powerful (even in the Euclidean space), all three supervised methods obtain promising results. Hence, our DCMH has a good generalization ability for intramodal retrieval, even though it is designed for cross-modal retrieval.

Training time. Finally, as Table 8 shows, we compare the training time with the baselines at 32 bits. Generally, all methods spend relatively little time on the low-dimensional datasets.

Table 6. Mean average precision (%) for the top 50 retrieved instances for image and text queries with nonlinear embedding for INRIA-Websearch and MIRFLickr. (Results in bold represent the best performance.)*

		INRIA-Websearch				MIRFLickr			
Number of bits		16	24	32	64	16	24	32	64
Image query	CVH	33.63	37.97	41.60	47.05	65.41	64.95	63.86	62.76
	CMFH	32.83	39.13	35.37	42.61	55.12	55.18	55.11	55.08
	IMH	27.89	30.95	31.66	35.51	63.64	63.06	64.69	62.94
	SCM_Seq	38.12	38.48	34.37	26.96	68.31	**71.85**	**71.69**	**72.57**
	DCMH	**49.48**	**52.76**	**55.48**	**51.00**	**68.68**	68.99	69.96	71.32
Text query	CVH	34.17	40.49	45.07	**53.61**	65.70	65.06	64.20	61.13
	CMFH	28.56	33.79	35.72	41.73	57.09	57.09	57.11	56.97
	IMH	30.25	33.88	36.02	41.82	63.76	63.18	64.22	63.43
	SCM_Seq	29.03	20.99	23.91	25.05	66.53	**70.38**	**72.24**	**70.15**
	DCMH	**52.49**	**56.64**	**61.13**	53.06	**68.59**	69.97	69.59	69.95

*All baseline methods adopt the same nonlinear embedding trick.

Table 7. Mean average precision (%) for the top 50 retrieved instances for image-to-image and text-to-text tasks for intramodal retrieval at 32 bits

Query	Dataset	Wiki	LabelMe	Wiki+	VOC+	INRIA-Websearch	MIRFLickr
Image-to-image	IMH	38.42	48.88	43.50	46.60	63.45	67.42
	SCM_Seq	32.07	68.21	48.49	74.82	66.12	70.37
	DCMH	41.23	78.21	55.38	85.49	68.32	70.78
Text-to-text	IMH	68.61	67.93	53.70	99.63	69.95	71.66
	SCM_Seq	56.36	88.95	78.93	99.21	71.34	81.71
	DCMH	71.31	90.49	85.23	99.99	82.18	83.53

Table 8. Training time (in seconds) on different datasets at 32 bits

Datasets	Wiki	LabelMe	Wiki+	VOC+	MIRFLickr
CVH	0.23	1.14	29.76	29.07	0.38
IMH	9.05	11.68	10.78	19.94	63.78
CMFH	0.16	0.42	33.37	2.93	0.99
SCM_Seq	0.20	67.88	862.72	798.67	41.01
DCMH_rat	0.64	0.60	0.63	1.45	2.12
DCMH	0.94	1.64	15.31	21.46	4.06

SCM_Seq always achieves the second best performance, but its training time significantly increases for high-dimensional data. With the help of discrete optimization, DCMH's performances can be further improved at the cost of additional time. Moreover, DCMH can easily adapt to high-dimensional datasets and large-scale datasets.

Heterogeneous hashing is a significant problem in social media, and we could further extend our current methods in a semisupervised manner to address this problem. This could help us get rid of expensive semantic labels. In addition, we could extend our methods to multimodal hashing to use heterogeneous information simultaneously for Web content retrieval. **MM**

Acknowledgments

This special issue is a collaboration between the 2016 IEEE International Symposium on Multimedia (ISM 2016) and *IEEE MultiMedia*. This article is an extended version of "Discrete Cross-Modal Hashing for Efficient Multimedia

Retrieval," presented at ISM 2016. Our work was supported in part by the National Natural Science Foundation of China (grant no. 61502073,61473289), the Foundation for Innovative Research Groups of the NSFC (grant no. 71421001), the Open Projects Program of National Laboratory of Pattern Recognition (grant no. 201407349), and the Strategic Priority Research Program of the Chinese Academy of Sciences (grant no. XDB02070000). This work was performed when Dekui Ma visited the National Laboratory of Pattern Recognition. Ma and Jian Liang are joint first authors for this article. Liang designed the research and is the corresponding author.

References

1. W. Liu et al., "Hashing with Graphs," presentation, Int'l Conf. Machine Learning (ICML-11), 2011; www.ee.columbia.edu/∼wliu/ICML11_agh_talk.pdf.
2. F. Shen et al., "Supervised Discrete Hashing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 37–45.
3. S. Kumar and U. Raghavendra, "Learning Hash Functions for Cross-View Similarity Search," *Proc. 20th Int'l Joint Conf. Artificial Intelligence* (IJCAI), 2011, pp. 1360–1365.
4. J. Liang et al., "Group-Invariant Cross-Modal Subspace Learning," *Proc. 25th Int'l Joint Conf. Artificial Intelligence* (IJCAI), 2016, pp. 1739–1735.
5. J. Liang et al., "Self-Paced Cross-Modal Subspace Matching," *Proc. 39th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, 2016, pp. 569–578.
6. G. Ding, Y. Guo, and J. Zhou, "Collective Matrix Factorization Hashing for Multimodal Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2014, pp. 2083–2090.
7. X. Shen et al., "Multi-View Latent Hashing for Efficient Multimedia Search," *Proc. 23rd ACM Int'l Conf. Multimedia*, 2015, pp. 831–834.
8. D. Zhang and W.J. Li, "Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization," *Proc. 28th AAAI Conf. Artificial Intelligence* (AAAI), 2014, pp. 2177–2183.
9. D. Ma et al., "Discrete Cross-Modal Hashing for Efficient Multimedia Retrieval," *IEEE Int'l Symp. Multimedia* (ISM), 2016, pp. 38–43.
10. D. Ma et al., "Frustratingly Easy Cross-Modal Hashing," *Proc. 2016 ACM on Multimedia Conf.*, 2016, pp. 237–241.
11. Z. Lin et al., "Semantics-Preserving Hashing for Cross-View Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015, pp. 3864–3872.
12. A. Sharma et al., "Generalized Multiview Analysis: A Discriminative Latent Space," *IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), 2012, pp. 2160–2167.
13. Y. Wei et al., "Modality-Dependent Cross-Media Retrieval," *ACM Trans. Intelligent Systems and Technology* (TIST), vol. 7, no. 4, 2016, article no. 57.
14. J. Song et al., "Inter-Media Hashing for Large-Scale Retrieval from Heterogeneous Data Sources," *Proc. 2013 ACM SIGMOD Int'l Conf. Management of Data*, 2013, pp. 785–796.

Dekui Ma is a third-year graduate student in the School of Information and Communication, Dalian University of Technology, where he studies under Xiangwei Kong. His research interests include multimedia retrieval and cross-modal hashing. Ma has a BE in information and communication engineering from Dalian University of Technology. Contact him at madk@mail.dlut.edu.cn.

Jian Liang is a PhD candidate at the National Laboratory of Pattern Recognition in the Chinese Academy of Sciences Institute of Automation (NLPR, CASIA). His research interests include machine learning, computer vision, and multimedia. Liang received a BE in electronic information and technology from Xi'an Jiaotong University. Contact him at jian.liang@nlpr.ia.ac.cn.

Ran He is a full professor at the National Laboratory of Pattern Recognition in the Chinese Academy of Sciences Institute of Automation (NLPR, CASIA). His research interests include information theoretic learning, pattern recognition, and computer vision. He has a PhD in pattern recognition and intelligent systems from NLPR, CASIA, and is a senior member of IEEE. Contact him at rhe@nlpr.ia.ac.cn.

Xiangwei Kong is a professor in the School of Information and Communication Engineering at Dalian University of Technology, China. Her research interests include digital image processing and recognition, multimedia information security, digital media forensics, image retrieval and mining, multisource information fusion, knowledge management, and business intelligence. Kong has PhD in management science and engineering from Dalian University of Technology. Contact her at kongxw@dlut.edu.cn.