Alternating Co-Quantization for Cross-modal Hashing

Go Irie Hiroyuki Arai Yukinobu Taniguchi
NTT Corporation
{irie.go, arai.hiroyuki, taniguchi.yukinobu}@lab.ntt.co.jp

Abstract

This paper addresses the problem of unsupervised learning of binary hash codes for efficient cross-modal retrieval. Many unimodal hashing studies have proven that both similarity preservation of data and maintenance of quantization quality are essential for improving retrieval performance with binary hash codes. However, most existing cross-modal hashing methods mainly have focused on the former, and the latter still remains almost untouched. We propose a method to minimize the binary quantization errors, which is tailored to cross-modal hashing. Our approach, named Alternating Co-Quantization (ACQ), alternately seeks binary quantizers for each modality space with the help of connections to other modality data so that they give minimal quantization errors while preserving data similarities. ACQ can be coupled with various existing cross-modal dimension reduction methods such as Canonical Correlation Analysis (CCA) and substantially boosts their retrieval performance in the Hamming space. Extensive experiments demonstrate that ACQ can outperform several state-of-the-art methods, even when it is combined with simple CCA.

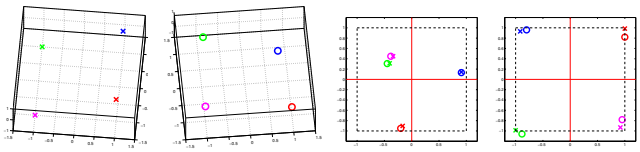
1. Introduction

Similarity-preserving hashing is a powerful indexing tool for efficient retrieval against massive databases. Especially, compact binary hashing has attracted much attention recently, because it can substantially reduce both query time and storage costs by encoding high-dimensional data into indexable compact binary codes [34, 33, 7]. Most research efforts have been made on *unimodal hashing*, where a query and database entries are both assumed to be in a homogeneous feature space [34, 33, 7, 25, 12, 9, 30, 13]. Meanwhile, in recent real-world scenarios such as Web or social media services, image is naturally surrounded by various side information sources such as tags, descriptions, and attributes [8, 14]. Moreover, some emerging topics arising at the intersection of computer vision and natural language processing such as automatic image description [17, 18] are based on matching between heterogeneous image-sentence data pairs. To enhance the efficiency of retrieval over such multimodal data sources, some recent studies have explored *cross-modal hashing* techniques [2, 23, 32, 28, 37, 4, 38, 39].

The goal of cross-modal hashing is to learn hash functions that give mappings from each of two (or more number of) different modality spaces to one common binary Hamming space, while preserving both intra- and inter-modal data similarities. Previous cross-modal hashing studies mainly focus on developing effective models to preserve data similarities. Yet in unimodal hashing studies, it has been proven that not only similarity preservation but also binary quantization quality is crucial to improve retrieval performance with binary hash codes [7, 6, 9, 13, 19, 20, 35, 3]. Nevertheless, to the best of our knowledge, there is no previous work that has focused on binary quantization for cross-modal hashing.

In this paper, we propose an approach to unsupervised learning of cross-modal binary hash codes, which aims at minimizing the binary quantization errors. One straightforward approach may be to use CCA-ITQ [7] which can be done by the following two-step procedure: first use Canonical Correlation Analysis (CCA) to find a common low-dimensional subspace where the inter-modal correlation is maximized, and then do Iterative Quantization (ITQ) to minimize the binary quantization errors by rotating the subspace[1]. This two-step approach actually works well and is even comparable to several state-of-the-art methods as shown later in our experiments. However, one shortcoming is that dimensions discarded in the CCA stage can never be recovered in the ITQ stage, which may be disadvantageous especially in cross-modal hashing scenarios – due to their inconsistency between data distributions of different modalities, good binary quantization may not always be achieved in a subspace chosen by CCA. One simple toy example is shown in Figure 1. Four data points are placed in each of two 3D ambient feature spaces whose intrinsic dimensions are 2 and 3, respectively (Figure 1(a,b)). The 2D embedding result by CCA-ITQ shown in Figure 1(c) gives almost perfect inter-modal matches. However, it fails to separate four data pairs as one per each quadrant, which is the ideal result in this setting. This is due to its suboptimal choice of subspaces by CCA in binary quantization.

[1]This procedure is exactly CCA-ITQ originally presented in [7], [7] uses this rather to learn binary hash codes in a supervised setting, though.



(a) Modal A (b) Modal B (c) CCA-ITQ (d) CCA-ACQ

Figure 1. A toy example. (a,b) Synthetic cross-modal data. Four data points are almost squarely arranged in each of two 3D spaces so that their intrinsic dimensions are 2 (in Modal A) and 3 (in Modal B), respectively. Points of the same colors are assumed to be relevant to each other. 2D embedding results by (c) CCA-ITQ and (d) our CCA-ACQ. Best viewed in color.

Motivated by these observations, our approach presented in this paper, named Alternating Co-Quantization (ACQ), aims at learning similarity-preserving binary quantizers by solving a joint optimization problem of subspace learning and binary quantization. The ACQ algorithm alternately updates the binary quantizers for each of multiple modality spaces, so as to minimize the distances between the inter-modal data pairs, corresponding binary hash codes, and vertices of the binary hypercube. The formulation of ACQ presented in this paper is based on a general framework of dimension reduction called Generalized Multiview Analysis (GMA) [29]. Therefore, ACQ can be coupled with many popular dimension reduction techniques which are in a specific class of quadratically constrained quadratic programming problems including CCA, Locality Preserving Projection (LPP) [11], Neighborhood Preserving Embedding (NPE) [10], and so on. Extensive experiments on three benchmark datasets demonstrate that our ACQ substantially boosts the retrieval quality of coupled base dimension reduction methods in the Hamming space and can outperform CCA-ITQ and several state-of-the-art methods even if it is with simple CCA.

Figure 1(d) shows the embedding result by our ACQ coupled with CCA (tagged as CCA-ACQ) for the toy dataset. As can be seen, our CCA-ACQ yields the ideal result where each of four pairs is allocated to each quadrant while maintaining reasonable inter-modal matches inside.

2. Related Work

We briefly review two relevant topics to this work, i.e., binary quantization and cross-modal hashing.

Binary Quantization. Problems of binary hash code learning often turn into difficult non-linear integer programming, due to binary constraints. Hence, most of the existing methods first relax their problems by ignoring the binary constraints to find some real-value low-dimensional embedding of data points, then take their sign to obtain binary hash codes [34, 33]. However, the results are often suffered from non-negligible binary quantization errors. Several methods have been proposed to reduce the binary quantization errors, which can be categorized into two major approaches: *multi-bit assignment* and *orthogonal quantization*. *Multi-bit assignment* aims at obtaining fine quantization results by assigning multiple bits to each dimension of the subspace. Double-Bit Quantization (DBQ) [19] assigns two bits to each dimension by using adaptively learned thresholds. Some other methods [21, 35] allow to use more bits. The representative method of *orthogonal quantization* is ITQ [7]. Starting with some similarity-preserving dimension reduction (typically PCA), ITQ introduces a new rotation matrix and iteratively updates it so that the projected data points fit to the binary hypercube vertices. Many extensions like Angular Quantization-based Binary Coding [6], Isotropic Hashing [20], and K-means Hashing [9] have also been proposed. Our ACQ is also inspired by ITQ. However, ACQ is tailored to the cross-modal hashing problem and aims at learning quantizers for multiple modality spaces simultaneously. Unlike existing methods, ACQ jointly optimizes similarity-preserving dimension reduction and binary quantization, which leads to significant performance improvements as shown later in our experiments.

Cross-modal Hashing. Most existing cross-modal hashing methods mainly focus on modeling intra- and inter-modal data similarities. To preserve inter-modal data correspondences, Cross-Modal Similarity Sensitive Hashing (CMSSH) [2] learns a set of hash functions in a boosting manner. Cross-View Hashing (CVH) [23] and Inter-Media Hashing (IMH) [32] are cross-modal extensions of Spectral Hashing [34] and aim at retaining both intra-modal data affinities and inter-modal data correlations in a common subspace. Predictable Dual-view Hashing (PDH) [28] refines initial CCA projections by learning linear SVMs and binary hash codes in a self-taught manner. Collective Matrix Factorization Hashing (CMFH) [4] assumes that a set of feature matrices of multiple modalities can be factorized into modal-specific matrices and a single common matrix, and learns modality-invariant hash codes so as to recover the common matrix. Other related methods are Co-Regularized Hashing [38] and Multimodal Latent Binary Embedding [39]. Unlike these, our ACQ aims at minimizing binary quantization errors.

3. Alternating Co-Quantization

We present our ACQ approach in this section. For simplicity, we hereafter assume that there are only two modalities. Note that our framework can readily be extended to cases of three or more modalities, as discussed later in Section 3.3.

Suppose we have data matrices of two different modalities, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$, where $\mathbf{x}_i \in \mathbb{R}^{d_x}$ and $\mathbf{y}_i \in \mathbb{R}^{d_y}$, $i = 1, \ldots, n$, and n is the number of data points. We assume that $\mathbf{x}_i$ and $\mathbf{y}_i$, $\forall i$, are related to each other in some sense (e.g., an image and an associated text). W.l.o.g., we assume that the mean vector over the data points in each modality space is $\mathbf{0}$. Given such data, our goal is to obtain similarity-preserving binary

quantizers q_x and q_y for these two modality spaces, which give mappings from each of $\mathbb{R}^{d_x}$ and $\mathbb{R}^{d_y}$ to a common c-dimensional Hamming space $\mathbb{H}^c$. In this paper, we consider the following specific form.

$$q_x(\mathbf{x}) = \mathrm{sgn}(A^\top \mathbf{x}), \quad q_y(\mathbf{y}) = \mathrm{sgn}(B^\top \mathbf{y}) \tag{1}$$

where $\mathrm{sgn}(\cdot)$ is the element-wise sign function, and $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_c] \in \mathbb{R}^{d_x \times c}$ and $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_c] \in \mathbb{R}^{d_y \times c}$ are parameters of the binary quantizers to be learned from data.

Our ACQ learns A and B by solving a joint optimization problem of similarity preservation and binary quantization. The basic structure of the objective function is

$$\mathcal{J}(A, B; X, Y) - \mathcal{Q}(A, B; X, Y) \tag{2}$$

where $\mathcal{J}$ and $\mathcal{Q}$ are *similarity preservation quality* function and *binary quantization error* function, respectively. We first give the details of $\mathcal{J}$ in Section 3.1 and $\mathcal{Q}$ in Section 3.2. Then we propose the total problem of ACQ and the algorithm to solve it in Section 3.3.

3.1. Similarity Preservation Quality

In order to find similarity-preserving compact binary codes, many previous formulations are grounded in some dimension reduction techniques (e.g., [23, 32, 28]). Similarly, we design our function $\mathcal{J}$ based on a dimension reduction framework. Specifically, we follow the GMA framework [29] which has the following form.

$$\mathcal{J}(A, B; X, Y) = \mathrm{tr}(A^\top C_{xx} A + B^\top C_{yy} B + 2\alpha A^\top C_{xy} B) \tag{3}$$

where C_{xx} and C_{yy} are symmetric definite matrices that give intra-modal correlations of X and Y, respectively, and C_{xy} is an inter-modal correlation matrix between X and Y. $\alpha > 0$ is a balancing hyperparameter. This is rewritten as a single matrix form as

$$\mathcal{J}(A, B; X, Y) = \mathrm{tr}\left(\begin{bmatrix} A \\ B \end{bmatrix}^\top \begin{bmatrix} C_{xx} & \alpha C_{xy} \\ \alpha C_{xy}^\top & C_{yy} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} \right). \tag{4}$$

By imposing some proper orthogonal constraints, e.g., $A^\top X X^\top A + \gamma B^\top Y Y^\top B = I$ (where $\gamma = \frac{\mathrm{tr}(XX^\top)}{\mathrm{tr}(YY^\top)}$), the problem of maximizing $\mathcal{J}$ w.r.t. A and B turns into a standard generalized eigenproblem which can be easily solved by using some eigensolver [29]. The solutions A and B are obtained by taking c eigenvectors corresponding to the top c largest eigenvalues, where their top d_x rows and the remaining d_y rows are for A and B, respectively.

Depending on how the matrices C_{xx}, C_{yy} and C_{xy} are defined, $\mathcal{J}$ becomes equivalent to various dimension reduction methods [29]. In this paper, we consider CCA and NPE [10] as our base methods.

CCA. If we set $C_{xx} = C_{yy} = 0$ and $C_{xy} = XY^\top$, then it is equivalent to the CCA objective. Many previous methods such as CVH [23], IMH [32], and PDH [28] are based on CCA to preserve inter-modal data similarities.

NPE. To incorporate intra-modal data similarity which is ignored in the CCA objective, we use NPE [10] as our second base method. NPE preserves local geometric structures of data in a subspace, which are essential for distance-based nearest neighbor search [36, 13]. Although the original NPE is only for unimodal dimension reduction, this can be readily extended to the cross-modal case [29], just by setting $C_{xx} = -X(I - W_x)^\top (I - W_x) X^\top$, $C_{yy} = -Y(I - W_y)^\top (I - W_y) Y^\top$, and $C_{xy} = XY^\top$. $W_x = [w_{ij}]_{i,j=1}^n$ here can be obtained by solving

$$\min_{W_x} \sum_{i=1}^n \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij} \mathbf{x}_j\|_2^2, \quad \text{s.t.} \quad \sum_{j \in N_i} w_{ij} = 1 \tag{5}$$

where N_i is the index set of K nearest neighbors of $\mathbf{x}_i$ [10], $w_{ij} = 0$ if $j \notin N_i$. W_y is also obtained by exactly the same procedure.

3.2. Binary Quantization Error

We minimize the binary quantization errors as done in [7]. Specifically, we aim at minimizing the distances between projected data points $A^\top X$ (resp. $B^\top Y$) and the corresponding binary hash codes $\mathrm{sgn}(A^\top X)$ (resp. $\mathrm{sgn}(B^\top Y)$). Formally, the total error can be given as

$$\|U - A^\top X\|_F^2 + \eta \|V - B^\top Y\|_F^2 \tag{6}$$

where $U \in \{\pm 1\}^{c \times n}$ and $V \in \{\pm 1\}^{c \times n}$ are the binary hash codes for X and Y, respectively. η is a balancing hyperparameter. (6) can be expanded as

$$\mathcal{Q}(A, B, U, V; X, Y) = -2\mathrm{tr}(UX^\top A + \eta V Y^\top B) + \mathrm{const.} \tag{7}$$

under the constraints of $A^\top X X^\top A = I$ and $B^\top Y Y^\top B = I$. So minimizing $\mathcal{Q}$ can be equivalent to maximizing the correlations between the binary hash codes U (resp. V) and the projected data points $A^\top X$ (resp. $B^\top Y$).

3.3. Total Problem & Algorithm

Now we propose the total problem of ACQ. By putting $\mathcal{J}$ (3) and $\mathcal{Q}$ (7) into together, the final formulation is

$$\max_{A,B,U,V} \ \mathrm{tr}(A^\top C_{xx} A + B^\top C_{yy} B + 2\alpha A^\top C_{xy} B) + \mathrm{tr}(2\lambda U X^\top A + 2\eta V Y^\top B) \tag{8}$$

$$\text{s.t. } A^\top X X^\top A + \gamma B^\top Y Y^\top B = I, \tag{9}$$

where λ is a balancing hyperparameter. This total problem is non-convex as is. Fortunately, each sub-problem for any one of the four matrices (A, B, U, and V) can be con-

vex with the other three fixed. Local optima thus can be obtained in an alternating optimization. Initializing A and B by some cross-modal dimension reduction method (such as CCA and CCA-ITQ) and setting $U = \mathrm{sgn}(A^\top X)$ and $V = \mathrm{sgn}(B^\top Y)$, our algorithm alternately updates the four matrices by repeating the following procedures (we found that taking a few subiterations within each modality gives reasonably stable results).

1. Update A (or B). Ignoring the terms including only fixed variables and relaxing the constraints into the objective as a penalty term as done in [33], the problem can be transformed as follows.

$$\max_A \ \mathrm{tr}(A^\top C_{xx} A + 2(\alpha B^\top C_{xy}^\top + \lambda U X^\top)A) \quad (10)$$
$$\text{s.t. } A^\top X X^\top A = I,$$
$$\Rightarrow$$
$$\max_A \ \mathrm{tr}(A^\top C_A A + 2 D_A A) \quad (11)$$

where $C_A = (C_{xx} - \beta X X^\top)$ and $D_A = (\alpha B^\top C_{xy}^\top + \lambda U X^\top)$. β is a penalty constant. Since C_A can always be symmetric definite for any C_{xx} defined in accordance with the GMA framework, this sub-problem is a standard quadratic programming problem which can be efficiently solved, for example, by using PCG method. Subsequently each column of A is ℓ^2-normalized. The update rule for B is also obtained in the same way.

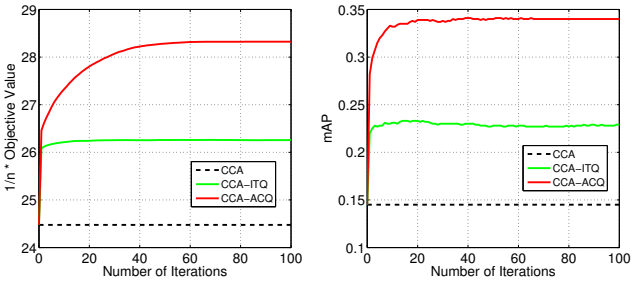
2. Update U (or V). In this case, the objective turns into the following simple form.

$$\max_{U \in \{\pm 1\}^{c \times n}} \ \mathrm{tr}(U X^\top A) \quad (12)$$

where its solution is $U = \mathrm{sgn}(A^\top X)$. The update rule for V is the same, i.e., $V = \mathrm{sgn}(B^\top Y)$.

ACQ is inspired by ITQ, but the algorithms are different. ITQ first applies some pre-fixed dimension reduction projections to the data points and then quantizes the projected data by an additional rotation matrix. In contrast, ACQ finds two separate binary quantizers for each of two modality spaces. Notably, once dimension reduction projections are fixed, it is difficult to find separate quantizers for each of two modality spaces. For instance, suppose we have CCA projections P and Q (that maximize the CCA objective $\mathrm{tr}(P^\top X Y^\top Q)$) and consider to find two separate quantizers R_x and R_y. In this case, the optimality of the original CCA objective may no longer hold, since $\mathrm{tr}(P^\top X Y^\top Q) \neq \mathrm{tr}(R_x P^\top X Y^\top Q R_y^\top)$ for any $R_x \neq R_y$. This fact is another motivation for us to consider the end-to-end joint optimization of binary quantizers.

Convergence Analysis. Figure 2(a) shows a typical behavior of the objective value (8) of our CCA-ACQ in the alternating iterations. The values by CCA and CCA-ITQ are shown in the figure as well. CCA-ACQ reasonably converges around 50 iterations and achieves much better objective values compared to CCA and CCA-ITQ. Figure 2(b) shows the corresponding behaviors of retrieval performance measured by mean Average Precision (mAP). Performance of CCA-ACQ tends to be improved as the number of iterations increases and is significantly better than CCA-ITQ after only a few iterations. ACQ does not need a number of iterations to achieve good performance. We typically use around 10 iterations in our experiments.



(a) Objective Value (b) mAP

Figure 2. Convergence analysis. (a) Objective value vs. the number of iterations. (b) mean Average Precision (mAP) vs. the number of iterations. Higher values mean better. These results are generated using 64-bit codes on Wiki dataset (details given later in Section 4).

Complexity Analysis. We analyze the computational complexity of the CCA-ACQ algorithm. We use $d = \max\{d_x, d_y\}$ for brevity. Time complexity for training is $O(nd^2 + tcnd + tcd^2)$ where t is the total number of iterations. It is linear in n and quadratic in d. Empirically, it takes 4.4 seconds when we train 64-bit codes on 10K data points of 128D through 30 iterations using MATLAB on a workstation with 2.6 GHz Intel Xeon CPU. Space complexity for training is $O(n(d+c)+cd)$. Once training is done, time and space complexities to generate a binary hash code for a new data point are both $O(cd)$ which is constant w.r.t. n and linear in d.

Extension. Analogous to GMA [29], our formulation can readily be extended to cases of three or more number of modalities. Suppose we have M modality data matrices denoted by $\{X_m\}$, $(m = 1, 2, \ldots, M)$. Then the total problem (8) can be extended as

$$\max_{\{A_m\},\{U_m\}} \ \sum_{l,m}^{M} \mathrm{tr}(\alpha_{l,m} A_l^\top C_{l,m} A_m) + \sum_{m}^{M} \mathrm{tr}(2\lambda_m U_m X_m^\top A_m) \quad (13)$$

$$\text{s.t. } \sum_{m}^{M} \gamma_m A_m^\top X_m X_m^\top A_m = I, \quad (14)$$

where A_m and U_m are the matrices of quantizer parameters and the binary codes for X_m, respectively. $C_{l,m}$ is a definite correlation matrix between X_l and X_m. $\alpha_{l,m}$ and λ_m are balancing hyperparameters and $\gamma_m = \frac{1}{\mathrm{tr}(X_m X_m^\top)}$. Deriva-

Table 1. Text retrieval performance by image query (I2T). mAP values for various code lengths.

	Wiki					a-Pascal					COCO				
# bits	$c = 16$	24	32	48	64	16	24	32	48	64	16	24	32	48	64
CVH	0.179	0.166	0.158	0.151	0.146	0.360	0.329	0.319	0.292	0.280	0.484	0.476	0.471	0.450	0.435
CMSSH	0.187	0.189	0.183	0.188	0.187	0.312	0.300	0.288	0.285	0.286	0.472	0.464	0.467	0.462	0.453
IMH	0.194	0.176	0.166	0.156	0.147	0.403	0.358	0.355	0.352	0.323	0.466	0.458	0.449	0.431	0.416
PDH	0.262	0.270	0.293	0.274	0.282	0.396	0.407	0.402	0.410	0.414	0.440	0.454	0.456	0.454	0.467
CMFH	0.253	0.282	0.282	0.287	0.313	0.483	0.469	0.473	0.482	0.504	0.486	0.501	0.517	0.536	0.545
CCA-Sign	0.181	0.166	0.159	0.150	0.145	0.377	0.354	0.340	0.316	0.299	0.483	0.477	0.469	0.450	0.435
CCA-DBQ	0.242	0.221	0.204	0.186	0.176	0.447	0.434	0.406	0.356	0.323	0.492	0.502	0.517	0.513	0.514
CCA-ITQ	0.243	0.228	0.228	0.227	0.231	0.476	0.411	0.426	0.395	0.465	0.516	0.526	0.533	0.538	0.547
NPE-Sign	0.224	0.208	0.199	0.187	0.170	0.394	0.365	0.348	0.324	0.320	0.493	0.487	0.476	0.466	0.439
NPE-DBQ	0.288	0.267	0.256	0.236	0.221	0.448	0.436	0.408	0.363	0.333	0.489	0.510	0.523	0.521	0.519
NPE-ITQ	0.284	0.274	0.261	0.263	0.265	0.462	0.419	0.432	0.395	0.438	0.516	0.528	0.532	0.540	0.543
CCA-ACQ	0.307	0.325	0.336	0.337	0.339	**0.516**	0.508	**0.511**	**0.499**	0.507	**0.531**	**0.536**	**0.544**	**0.552**	0.554
NPE-ACQ	**0.322**	**0.338**	**0.351**	**0.349**	**0.352**	0.497	**0.512**	0.484	0.492	**0.522**	0.520	0.535	0.543	0.549	**0.555**

Table 2. Image retrieval performance by text query (T2I). mAP values for various code lengths.

	Wiki					a-Pascal					COCO				
# bits	$c = 16$	24	32	48	64	16	24	32	48	64	16	24	32	48	64
CVH	0.170	0.161	0.153	0.147	0.143	0.340	0.317	0.315	0.304	0.289	0.480	0.472	0.467	0.446	0.432
CMSSH	0.177	0.176	0.172	0.179	0.181	0.293	0.296	0.290	0.286	0.288	0.465	0.461	0.454	0.454	0.446
IMH	0.182	0.166	0.161	0.151	0.144	0.291	0.318	0.328	0.355	0.354	0.454	0.450	0.442	0.424	0.410
PDH	0.242	0.247	0.271	0.249	0.265	0.364	0.367	0.362	0.371	0.370	0.439	0.450	0.453	0.453	0.467
CMFH	0.237	0.250	0.262	0.276	0.280	0.451	0.441	0.454	0.462	**0.477**	0.479	0.501	0.519	0.536	0.547
CCA-Sign	0.175	0.161	0.155	0.148	0.143	0.431	0.425	0.415	0.382	0.362	0.477	0.471	0.465	0.446	0.432
CCA-DBQ	0.223	0.206	0.194	0.178	0.170	0.447	0.434	0.406	0.356	0.323	0.493	0.503	0.515	0.511	0.510
CCA-ITQ	0.223	0.215	0.211	0.211	0.217	0.463	0.423	0.433	0.418	0.443	0.518	0.531	0.536	0.542	0.550
NPE-Sign	0.211	0.187	0.176	0.164	0.151	0.422	0.420	0.410	0.377	0.364	0.492	0.487	0.479	0.469	0.442
NPE-DBQ	0.271	0.252	0.244	0.230	0.218	0.448	0.436	0.408	0.363	0.333	0.489	0.512	0.521	0.518	0.516
NPE-ITQ	0.253	0.252	0.253	0.239	0.240	0.459	0.425	0.405	0.415	0.410	0.515	0.530	0.533	0.538	0.545
CCA-ACQ	0.295	0.290	0.298	0.295	0.303	**0.474**	0.463	0.475	**0.472**	0.472	0.520	**0.543**	0.543	**0.556**	**0.562**
NPE-ACQ	**0.298**	**0.311**	**0.312**	**0.304**	**0.309**	0.451	**0.470**	**0.476**	0.469	**0.477**	**0.521**	0.540	**0.546**	0.554	0.561

tion of alternating update rules for each of A_m and $U_m, \forall m$, is straightforward.

4. Experiments

We experimentally analyze performance of our ACQ in the two common tasks of cross-modal retrieval: one is text retrieval by image query (I2T) and the other is image retrieval by text query (T2I). We use CCA and NPE for the base methods of ACQ (see Section 3.1), which are denoted as CCA-ACQ and NPE-ACQ, respectively. The hyperparameters are tuned by standard parallel grid-search on a subset of training data. We compare our ACQ with three existing quantization methods: Sign (just taking the sign of CCA or NPE embedding), DBQ [19], and ITQ [7]. Furthermore, we also evaluate five state-of-the-art cross-modal hashing methods including CVH [23], CMSSH [2], IMH [32], PDH [28], and CMFH [4] by using Matlab codes provided by each author group. Their hyperparameters are carefully tuned for each experiment.

We follow the common evaluation protocol for cross-modal hashing [4, 28, 32]. Retrieval is performed in the Hamming ranking manner [7], where retrieved data points are sorted in ascending order of their Hamming distances from the query. As in [4, 28, 32], retrieval performance is measured by mean Average Precision (mAP) values and the retrieval is judged as successful if and only if the semantic label of a retrieved data point is the same as that of the query.

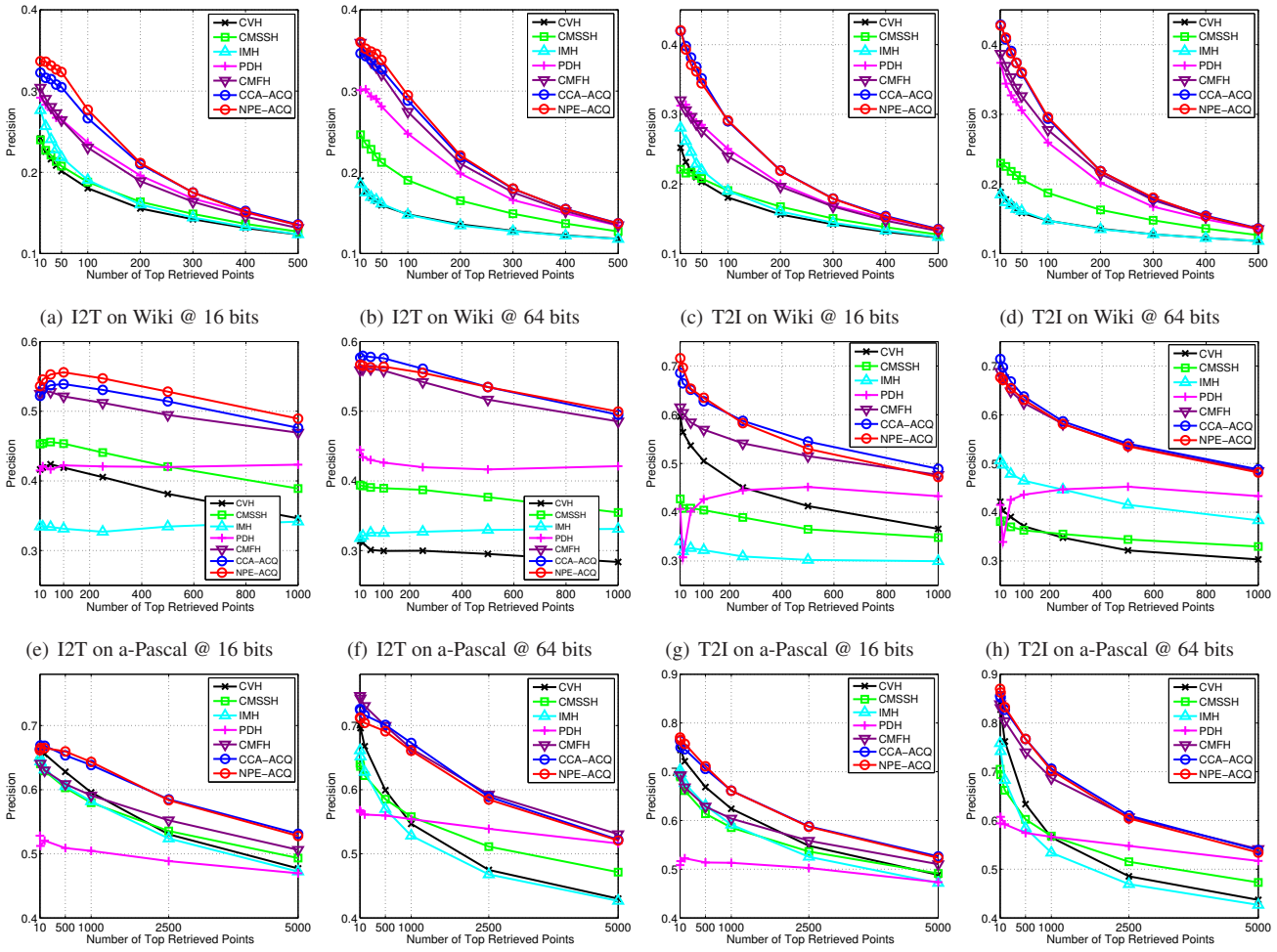
4.1. Datasets

The following three multimodal benchmark datasets are used in our experiments.

Wiki[2] [27]. This dataset contains $2,866$ articles collected from Wikipedia Featured Articles. Each article consists of a pair of an image and a text description which is categorized into 10 semantic topic classes. Our image feature is extracted by using the Caffe implementation[3] of a Convolutional Neural Network (CNN) called AlexNet [22]. Specifically, we first extract $4,096$D activation features from its fc6 layer, then reduce their dimension to 128D by PCA as done in [1]. For text feature, we use 100D skip-gram word vectors [26] learned by word2vec[4] and compute a mean vector of the word vectors of the words appear in each text description. Following its standard data split, we use 693 documents for testing and the other $2,173$ for training.

a-Pascal[5] [5]. This dataset contains $12,695$ images of 20 categories of objects. We use multiple handcrafted image

[2] http://www.svcl.ucsd.edu/projects/crossmodal/
[3] http://caffe.berkeleyvision.org/
[4] https://code.google.com/p/word2vec/
[5] http://vision.cs.uiuc.edu/attributes/



(a) I2T on Wiki @ 16 bits (b) I2T on Wiki @ 64 bits (c) T2I on Wiki @ 16 bits (d) T2I on Wiki @ 64 bits

(e) I2T on a-Pascal @ 16 bits (f) I2T on a-Pascal @ 64 bits (g) T2I on a-Pascal @ 16 bits (h) T2I on a-Pascal @ 64 bits

(i) I2T on COCO @ 16 bits (j) I2T on COCO @ 64 bits (k) T2I on COCO @ 16 bits (l) T2I on COCO @ 64 bits

Figure 3. Precision values at various numbers of top retrieved data points. Results for 16-bit and 64-bit codes are shown.

features [5]: we first extract a long vector by concatenating the features of texture, HOG, edge, and color, and then reduce their dimension to 128D by PCA. Each image is associated with a 64D binary attribute vector each of which indicates a part or some semantic property of an object (e.g., *leg*, *wing*, and *2D-boxy*). We use these binary attribute vectors as our text features. In this dataset, we randomly sample $1,000$ images for query and use the rest to construct training and database sets.

COCO[6] [24]. Microsoft COCO v2014.1 is a large-scale image dataset that contains $123,558$ images of 80 categories of objects. Each image is associated with five short sentences describing its content. In our experiments, we keep only the first sentence for each image. We use the same features as Wiki, i.e., AlexNet activation features and skip-gram word vectors for image and text features, respectively. We randomly sample $1,000$ images for query and use the rest to construct training and database sets. In this dataset, each image is allowed to have multiple labels, so we judge the retrieval is successful if a query and a retrieved image share at least one common label.

[6] http://mscoco.org/

4.2. Results

For all the three datasets, mAP values for various code lengths are reported in Table 1 (for the I2T task) and Table 2 (for the T2I task), and precision values at various numbers of top retrieved data points are shown in Figure 3.

Results on Wiki. First, as can be seen in Tables 1 and 2, we found that NPE-ACQ achieves the best mAP values in all the cases, and the second best is by CCA-ACQ which uses simple CCA as its base method. The maximum gains of NPE-ACQ over the third best method, CMFH, reaches 22.8% on I2T and 24.4% on T2I. These results clearly demonstrate its strong effectiveness of the proposed ACQ for cross-modal hashing. Second, CCA-ITQ and CCA-DBQ (resp. NPE-ITQ and NPE-DBQ) always improve CCA-Sign (resp. NPE-Sign). CCA-ACQ and NPE-ACQ further boost their performance. Interestingly, CCA/NPE-

ITQ and CCA/NPE-DBQ are already better than several baselines in some cases. These results suggest that maintaining binary quantization quality is essential for cross-modal hashing, as in the cases of unimodal hashing. Third, CMFH and PDH tend to get better mAP values as the number of bits increases. Conversely, some methods such as CVH, IMH, CCA-Sign, and NPE-Sign clearly decrease. As discussed in [4, 28], this may be due to orthogonal eigendecomposition involved in these methods, which enforces to pick uninformative low-variance dimensions. Notably, orthogonal quantization can mitigate this harmful effect by rotating data distributions so as to be nearly isotropic [7, 15]. Thanks to this property, CCA-ACQ and NPE-ACQ can yield much better mAP values for long codes. Finally, as shown in Figure 3(a-d), CCA-ACQ and NPE-ACQ are always better than all the baselines in precision values for top retrieved data points.

Results on a-Pascal. In mAP values shown in Tables 1 and 2, CCA-ACQ or NPE-ACQ is the best in all the cases, which emphasizes the effectiveness of ACQ. Unlike Wiki, behaviors of mAP values to the number of bits are not very monotonous in most of the methods. This may be because the text feature of this dataset is binary attribute whose distributions are difficult to be captured due to its high sparsity. CMFH, which learns hash codes so as to recover the latent inter-modal relational matrix rather than data distributions, yields relatively better performance on this dataset and is comparable with NPE-ACQ for 64-bit codes on the T2I task. In precision for top retrieved data points shown in Figure 3(e-h), CCA-ACQ and NPE-ACQ are better than the other state-of-the-art methods in most cases.

Results on COCO. Even for this larger-scale dataset, overall tendency is quite similar to the other two datasets; again, CCA-ACQ or NPE-ACQ shows the best mAP values in all the settings. These results further emphasize the importance of binary quantization and its strong effectiveness of the proposed ACQ approach.

Radius Search Performance. Another popular retrieval procedure is radius search with hash lookup tables, which retrieves data points lie in buckets within some small Hamming distances from a query [7]. Table 3 shows the comparative results at Hamming radius within 2. All the methods yield reasonable performance for short codes. However, CVH, CMSSH, and IMH fail to find data points as the number of bits increases. Meanwhile, CCA-ACQ, NPE-ACQ, and PDH successfully retrieve data points for even 64-bit codes, and CCA-ACQ and NPE-ACQ tend to get better performance compared to PDH.

Image-to-Image Search Performance. Several studies in the literature [32, 28] have proven that cross-modal hashing can also improve unimodal search performance by leveraging semantic information carried by text modality data. We report comparative results on the task of image retrieval by image query (I2I) in Table 4. As can be seen, CCA-ACQ and NPE-ACQ are always better than the natural baseline, PCA-ITQ [7], and achieve the best mAP values in all the cases. As for the other methods, CMFH and PDH show relatively better performance compared to the other existing methods. Note that CCA-ITQ and NPE-ITQ do already a good job and are competitive to CMFH or PDH in many cases. These results suggest that binary quantization is also crucial for improving unimodal search performance by cross-modal hashing methods.

Table 3. Radius search performance by hash lookup within Hamming radius 2. Precision values for various code lengths. '-' means unsuccessful retrieval, i.e., no item is found in the setting.

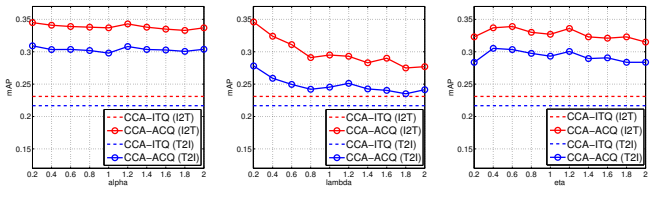
	I2T on Wiki			T2I on Wiki		
# bits	$c = 16$	32	64	16	32	64
CVH	0.335	-	-	0.335	-	-
CMSSH	0.265	0.000	-	0.241	-	-
IMH	**0.387**	-	-	0.387	-	-
PDH	0.208	0.303	0.269	0.208	0.303	0.269
CMFH	0.335	-	-	0.426	0.457	-
CCA-ACQ	0.379	**0.602**	0.689	**0.446**	0.640	**0.670**
NPE-ACQ	0.367	0.519	**0.697**	0.419	**0.654**	0.647
	I2T on a-Pascal			T2I on a-Pascal		
# bits	$c = 16$	32	64	16	32	64
CVH	0.546	0.400	-	0.564	0.500	-
CMSSH	0.460	0.000	-	0.519	-	-
IMH	0.331	-	-	0.257	0.027	-
PDH	0.309	0.390	0.511	0.314	0.403	0.516
CMFH	0.730	0.748	-	**0.777**	0.722	-
CCA-ACQ	0.765	0.895	**0.993**	0.768	**0.869**	0.909
NPE-ACQ	**0.785**	**0.896**	0.975	0.752	0.841	**0.944**
	I2T on COCO			T2I on COCO		
# bits	$c = 16$	32	64	16	32	64
CVH	**0.736**	0.928	-	0.734	0.933	-
CMSSH	0.627	0.763	-	0.635	0.774	-
IMH	0.709	0.791	-	0.696	0.925	-
PDH	0.456	0.438	0.504	0.462	0.439	0.506
CMFH	0.638	0.846	0.727	0.669	0.845	0.909
CCA-ACQ	0.701	0.903	0.996	0.715	0.924	0.992
NPE-ACQ	0.729	**0.939**	**0.999**	**0.734**	**0.956**	**0.999**

Parameter Sensitivity. We empirically analyze the sensitivity of retrieval performance of CCA-ACQ to the three hyperparameters α, λ, and η (see (8)) using Wiki (similar tendencies are observed on the other datasets). The results are shown in Figure 4. First, CCA-ACQ is better than CCA-ITQ for wide ranges of these values. Hence precise tuning may not always be necessary to achieve satisfactory performance. Second, performance is rather sensitive to λ and η which control the tradeoff between similarity preservation and binary quantization in each modality space. This is also an evidence that joint optimization is crucial to improve cross-modal hashing.

Image Description Performance. Lastly, we apply the CCA-ACQ binary codes to the retrieval-based image description task on COCO. We follow the common evaluation protocol for this task [18, 17]. 5K images and corresponding 25K sentences are used for testing (query and database) and the rest are used for training. We report (i) Recall@K (R@K) which is the fraction of queries for which a correct

Table 4. Image-to-Image (I2I) unimodal search performance. mAP values for various code lengths.

	Wiki					a-Pascal					COCO				
# bits	$c = 16$	24	32	48	64	16	24	32	48	64	16	24	32	48	64
PCA-ITQ	0.173	0.166	0.179	0.168	0.164	0.282	0.284	0.283	0.282	0.283	0.437	0.448	0.449	0.454	0.456
CVH	0.152	0.144	0.142	0.139	0.137	0.315	0.309	0.312	0.312	0.308	0.443	0.433	0.424	0.408	0.397
CMSSH	0.168	0.165	0.159	0.165	0.168	0.310	0.309	0.304	0.303	0.295	0.441	0.433	0.437	0.439	0.436
IMH	0.164	0.159	0.153	0.146	0.143	0.298	0.286	0.279	0.272	0.267	0.426	0.418	0.411	0.397	0.388
PDH	0.199	0.203	0.219	0.203	0.215	0.324	0.329	0.328	0.329	0.328	0.436	0.444	0.449	0.447	0.451
CMFH	0.195	0.219	0.221	0.226	0.225	0.390	0.391	0.394	0.395	0.400	0.462	0.476	0.484	0.497	0.505
CCA-DBQ	0.186	0.174	0.166	0.155	0.152	0.351	0.338	0.327	0.313	0.302	0.465	0.466	0.471	0.462	0.456
CCA-ITQ	0.190	0.173	0.181	0.179	0.177	0.356	0.347	0.349	0.342	0.340	0.477	0.481	0.485	0.490	0.494
NPE-DBQ	0.206	0.204	0.202	0.198	0.191	0.343	0.340	0.331	0.321	0.312	0.460	0.470	0.474	0.465	0.460
NPE-ITQ	0.224	0.216	0.222	0.226	0.229	0.352	0.344	0.342	0.336	0.336	0.476	0.482	0.484	0.487	0.490
CCA-ACQ	0.225	0.221	0.232	0.230	0.234	0.395	**0.411**	**0.412**	**0.417**	**0.422**	**0.483**	**0.500**	**0.504**	**0.515**	**0.520**
NPE-ACQ	**0.238**	**0.237**	**0.248**	**0.241**	**0.245**	**0.396**	0.401	0.404	0.402	0.404	0.478	0.483	0.492	0.502	0.509



(a) α (b) λ (c) η

Figure 4. Parameter sensitivity on Wiki. mAP values with various parameter settings of (a) α, (b) λ, and (c) η. See (8) for details of the parameters.

item is found among the top K results; and (ii) the median rank (Med) of the first retrieved ground truth item. If there are multiple items having an equal Hamming distance, we assign their mean rank to all of them. In this evaluation, we use $4,096$D CNN activation features extracted from the fc6 layer of VGGNet [31] and 300D skip-gram word vectors for image and text representations, respectively. We found that early stopping of training gives better performance for this task, so we use a fewer number of iterations for training.

Table 5 shows the results. It is shown that CCA-ACQ outperforms CCA-ITQ for this task. Not surprisingly, CCA-ACQ is worse than the recent strong image description methods [18, 17, 16]. This is because CCA-ACQ uses compact binary codes, while these methods rely on much higher-dimensional real-value vectors learned by powerful neural network models of fine-tuned or extended versions of Fisher kernels for sentence embedding. One advantage of using compact binary codes is efficient retrieval, which is especially beneficial to larger-scale applications.

5. Conclusions

We have proposed a novel approach to cross-modal hashing, named Alternating Co-Quantization (ACQ). As the first work that mainly considers binary quantization for cross-modal hashing, we have brought several new insights to the community. First, similar to the cases of unimodal hashing, minimizing the binary quantization errors is important to improve cross-modal hashing performance. In particular, even simple CCA has already achieved highly comparable

Table 5. Image description performance on COCO.

Image Annotation				
Method	R@1	R@5	R@10	Med
CCA-ITQ 64 bits	2.3	10.9	18.5	52
CCA-ITQ 128 bits	3.5	15.9	25.2	34
CCA-ITQ 256 bits	4.1	16.7	25.9	39
CCA-ACQ 64 bits	2.6	12.9	21.5	43.5
CCA-ACQ 128 bits	4.4	18.4	28.4	32
CCA-ACQ 256 bits	**5.2**	**19.6**	**30.1**	**30**
CCA (real-value) 64D	6.5	19.8	30.2	29
CCA (real-value) 128D	8.5	24.5	35.9	22
CCA (real-value) 256D	7.9	24.4	35.8	22
BRNNv1 [16]	11.8	32.5	45.4	12.2
BRNN [17]	16.5	39.2	52.0	9
GMM+HGLMM [18]	17.3	39.0	50.2	10
Image Search				
Method	R@1	R@5	R@10	Med
CCA-ITQ 64 bits	2.1	10.6	18.7	45
CCA-ITQ 128 bits	3.5	13.9	22.8	41
CCA-ITQ 256 bits	3.7	13.5	21.5	79
CCA-ACQ 64 bits	2.2	10.9	18.8	45
CCA-ACQ 128 bits	3.6	14.3	23.4	**36.5**
CCA-ACQ 256 bits	**4.1**	**15.4**	**24.1**	39.5
CCA (real-value) 64D	5.6	17.2	26.5	32
CCA (real-value) 128D	6.4	19.6	29.9	28
CCA (real-value) 256D	5.7	19.0	28.9	33
BRNNv1 [16]	8.9	24.9	36.3	19.5
BRNN [17]	10.7	29.6	42.2	14
GMM+HGLMM [18]	10.8	28.3	40.1	17

performance with several state-of-the-art methods, when it is coupled with ITQ. Second, we have shown that joint optimization of similarity preservation and binary quantization is crucial for improving cross-modal hashing quality. Our ACQ has yielded much better retrieval performance compared with ITQ, when they are combined with CCA or NPE. Finally, we have empirically demonstrated that CCA-ACQ and NPE-ACQ can outperform some recent state-of-the-art cross-modal hashing methods.

One limitation of the current ACQ formulation is that, it assumes the GMA framework [29] for its base methods, hence it cannot be coupled with any dimension reduction method outside of the form (3). Even though GMA covers many dimension reduction methods, extending the formulation so that can be combined with more various types of base methods will be an interesting future direction.

References

[1] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 5

[2] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, 2010. 1, 2, 5

[3] C. Deng, H. Deng, X. Liu, and Y. Yuan. Adaptive multi-bit quantization for hashing. *Neurocomputing*, 151(1):319–326, 2015. 1

[4] G. Ding, Y. Guo, and J. Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, 2014. 1, 2, 5, 7

[5] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 5, 6

[6] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik. Angular quantization-based binary codes for fast similarity search. In *NIPS*, 2012. 1, 2

[7] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. PAMI*, 35(12):2916–2929, 2013. 1, 2, 3, 5, 7

[8] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *CVPR*, 2010. 1

[9] K. He, F. Wen, and J. Sun. K-means hashing: an affinity-preserving quantization method for learning binary compact codes. In *CVPR*, 2013. 1, 2

[10] X. He, D. Cai, S. Yan, and H. J. Zhang. Neighborhood preserving embedding. In *ICCV*, 2005. 2, 3

[11] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003. 2

[12] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon. Spherical hashing. In *CVPR*, 2012. 1

[13] G. Irie, Z. Li, X.-M. Wu, and S.-F. Chang. Locally linear hashing for extractracting non-linear manifolds. In *CVPR*, 2014. 1, 3

[14] G. Irie, D. Liu, Z. Li, and S.-F. Chang. A bayesian approach to multimodal visual dictionary learning. In *CVPR*, 2013. 1

[15] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. PAMI*, 33(1):117–128, 2011. 7

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *arXiv preprint arXiv:1412.2306v1*, 2014. 8

[17] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 7, 8

[18] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 1, 7, 8

[19] W. Kong and W.-J. Li. Double-bit quantization for hashing. In *AAAI*, 2012. 1, 2, 5

[20] W. Kong and W.-J. Li. Isotropic hashing. In *NIPS*, 2012. 1, 2

[21] W. Kong, W.-J. Li, and M. Guo. Manhattan hashing for large-scale image retrieval. In *SIGIR*, 2012. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5

[23] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, 2011. 1, 2, 3, 5

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[25] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *ICML*, 2011. 1

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 5

[27] N. Rasiwasia, J. Costa Pereira, E. Coviello, and G. Doyle. Learning hash functions for cross-view similarity search. In *ACM Multimedia*, 2010. 5

[28] M. Rastegari, J. Choi, S. Fakhraei, H. Daumé III, and L. S. Davis. Predictable dual-view hashing. In *ICML*, 2013. 1, 2, 3, 5, 7

[29] A. Sharma, A. Kumar, and H. Daumé III. Generalized multiview analysis: A discriminative latent space. In *CVPR*, 2012. 2, 3, 4, 8

[30] F. Shen, C. Shen, Q. Shi, A. van den Hengel, and Z. Tang. Inductive hashing on manifolds. In *CVPR*, 2013. 1

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014. 8

[32] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Intermedia hashing for large-scale retrieval from heterogenous data sources. In *SIGMOD*, 2013. 1, 2, 3, 5, 7

[33] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *IEEE Trans. PAMI*, 34:2393–2406, 2012. 1, 2, 4

[34] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008. 1, 2

[35] C. Xiong, W. Chen, G. Chen, D. M. Johnson, and J. J. Corso. Adaptive quantization for hashing: An information-based approach to learning binary codes. In *SDM*, 2014. 1, 2

[36] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *PAMI*, 34(4):723–742, 2012. 3

[37] D. Zhang and W.-J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, 2014. 1

[38] Y. Zhen and D.-Y. Yeung. Co-regularized hashing for multimodal data. In *NIPS*, 2012. 1, 2

[39] Y. Zhen and D.-Y. Yeung. A probabilistic model for multimodal hash function learning. In *KDD*, 2012. 1, 2