Semi-Relaxation Supervised Hashing for Cross-Modal Retrieval

Peng-Fei Zhang
School of Computer Science and Technology, Shandong University
China
mima.zpf@gmail.com

Chuan-Xiang Li
School of Computer Science and Technology, Shandong University
China
lcx543576178@gmail.com

Meng-Yuan Liu
School of Computer Science and Technology, Shandong University
China
mayoliu.sdu@gmail.com

Liqiang Nie*
School of Computer Science and Technology, Shandong University
Jinan, China
nieliqiang@gmail.com

Xin-Shun Xu*
School of Computer Science and Technology, Shandong University
Jinan, China
xuxinshun@sdu.edu.cn

ABSTRACT

Recently, some cross-modal hashing methods have been devised for cross-modal search task. Essentially, given a similarity matrix, most of these methods tackle a discrete optimization problem by separating it into two stages, i.e., first relaxing the binary constraints and finding a solution of the relaxed optimization problem, then quantizing the solution to obtain the binary codes. This scheme will generate large quantization error. Some discrete optimization methods have been proposed to tackle this; however, the generation of the binary codes is independent of the features in the original space, which makes it not robust to noise. To consider these problems, in this paper, we propose a novel supervised cross-modal hashing method—Semi-Relaxation Supervised Hashing (SRSH). It can learn the hash functions and the binary codes simultaneously. At the same time, to tackle the optimization problem, it relaxes a part of binary constraints, instead of all of them, by introducing an intermediate representation variable. By doing this, the quantization error can be reduced and the optimization problem can also be easily solved by an iterative algorithm proposed in this paper. Extensive experimental results on three benchmark datasets demonstrate that SRSH can obtain competitive results and outperform state-of-the-art unsupervised and supervised cross-modal hashing methods.

CCS CONCEPTS

• **Computing methodologies → Learning paradigms**; • **Information systems → Multimedia and multimodal retrieval**;

KEYWORDS

Multimodal, Hashing, Cross-Modal Search, Approximate Nearest Neighbor Search

1 INTRODUCTION

In many applications, we need to search a database to find the nearest neighbors of a query. However, it becomes time-consuming and infeasible for large-scale data. Therefore, in these years, approximate nearest neighbor (ANN) search has attracted much attention in many fields including information retrieval, data mining and computer vision [20, 21, 27, 29]. Especially, hashing-based ANN search technology is becoming more and more attractive due to its fast query speed and low storage cost.

In the last decade, many hashing-based ANN search methods have been proposed and obtained promising performance. Most of the pioneer efforts focus on the search task of the single-modal scenario, e.g., Text-to-Text or Image-to-Image. In many cases, data may have multiple modalities. To make full use of the information contained in multiple modalities, various multimodal hashing methods have been devised. However, in real applications, it is usually difficult to make all data samples have all modalities; for example, the query only has one modality. To tackle this problem, cross-modal search is becoming increasingly attractive, through which users can get the results with different modalities by submitting a query with some type of modality [33, 34]. Correspondingly, cross-modal hashing methods have attracted more and more attention in recent years, and various cross-modal hashing methods have been proposed.

Given a similarity matrix, these methods map the data samples into binary codes while preserving the similarity by optimizing a discrete optimization problem which is hard to solve. To tackle the optimization problem, many methods separate it into two independent stages, i.e., first relaxing the binary constraints and finding a solution of the relaxed optimization problem, then quantizing the solution to obtain the binary codes. Such scheme will generate large quantization error. In addition, some discrete optimization methods have been proposed to optimize the binary codes directly;

*Corresponding author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.
© 2017 ACM. ISBN 978-1-4503-4906-2/17/10...$15.00
DOI: http://dx.doi.org/10.1145/3123266.3123320

however, the generation of the binary codes is independent of the features in the original space, which makes it not robust to noise.

To consider these problems, in this paper, we propose a novel supervised cross-modal hashing method, i.e., Semi-Relaxation Supervised Hashing (SRSH). It can learn the hash functions and the binary codes simultaneously. At the same time, to tackle the optimization problem, it relaxes a part of binary constraints, instead of all of them, by introducing an intermediate representation variable. Moreover, the hashing functions can also be learnt simultaneously. By doing this, the quantization error can be reduced and the optimization problem can also be easily solved by an iterative algorithm proposed in this paper. Extensive experimental results on three benchmark datasets including Wiki, MIRFlickr-25K, and NUSWIDE demonstrate that SRSH can obtain competitive results and outperform state-of-the-art unsupervised and supervised cross-modal hashing methods.

The contributions of this work are summarized as follows:

- A novel supervised cross-modal hashing method is proposed, which can reduce the quantization error by relaxing only a part of binary constraints.
- An iterative algorithm is proposed to solve the optimization problem of the proposed hashing method.
- The proposed method obtains competitive results compared with state-of-the-art hashing methods for cross-modal search task.

The rest of this paper is organized as follows. The related work is discussed in Section 2. Section 3 introduces the proposed SRSH model including the framework, optimization algorithm and its extensions to out-of-sample data and more modalities. Section 4 presents the experimental results and some analysis on three benchmark datasets. Finally, Section 5 concludes this paper.

2 RELATED WORK

Existing hashing methods can be divided into data-independent and data-dependent ones. The former generates hash functions by random projections without considering specific data. For example, Locality-Sensitive Hashing (LSH) [6] is one of the most popular data-independent ones, which has been applied to many applications. Usually, LSH needs long binary codes to obtain good performance, which limits its scalability. The data-dependent one generates hash functions by considering specific data and usually could obtain compact binary codes, which can be further classified into unsupervised and supervised ones according to whether the supervised information, e.g., semantics labels/tags, is used during the learning of hash functions or binary codes. As its word implies, unsupervised ones map data into a Hamming space without utilizing the semantic information. Typical examples are Spectral Hashing (SH) [31], Iterative Quantization (ITQ) [7], Isotropic Hashing (IsoHash) [12], Discrete Graph Hashing (DGH) [17], Linear Distance Preserving Hashing [30] and Scalable Graph Hashing (SGH) [9]. Different from the unsupervised methods, supervised ones learn hash functions/binary codes by making full use of supervised information. Many supervised hashing methods have been proposed, such as Sequential Projection Learning for Hashing (SPLH) [28], Minimal Loss Hashing (MLH) [22], Supervised Hashing with Kernels (KSH) [18], LDAHash [26], Two-Step Hashing (TSH) [15], FastHash [14], Binary Optimized Hashing [4], and Supervised Discrete Hashing (SDH) [24], etc.

Cross-modal hashing methods need to consider the inter- and intra-modality relatedness; therefore, most of them are data-dependent methods, which can also be classified into unsupervised and supervised ones. Unsupervised ones map data samples into binary codes by exploiting the inter- and intra-modality relatedness of the given data without utilizing supervised information. Inter-Media Hashing (IMH) [25], Linear Cross-Modal Hashing (LCMH) [41], Latent Semantic Sparse Hashing (LSSH) [40], Collective Matrix Factorization Hashing (CMFH) [5], and Composite Correlation Quantization (CCQ) [19] are all representative cross-modal hashing methods. IMH explores the correlations among multiple modalities by introducing inter- and intra-media consistency to discover a common Hamming space, and uses a linear regression with regularization model to learn modality-specific hash functions. To solve the scalability issue for large-scale data, LCMH first partitions the training data of each modality into k clusters by applying a linear-time clustering method, and then represents each training data points using its distances to the k clusters' centroids to achieve a linear-time complexity with respect to the training data size in the training phase. LSSH employs sparse coding to capture the salient structures of the images and matrix factorization to learn the latent concepts from texts; thereafter, the learnt latent semantic features are mapped to a joint abstraction space. CMFH learns unified hash codes by collective matrix factorization with latent factor model from different modalities, which can not only supports cross-modal search but also increases the search accuracy by merging multiple modalities/views. CCQ first transforms different modalities into an isomorphic latent space, and then learns the composite quantizers that convert the isomorphic latent features into compact binary codes.

By comparison, supervised cross-modal hashing methods can not only exploiting the inter- and intra-modality relatedness, but also leverage supervised information. Quite a few such methods have been proposed. For example, Cross Modality Similarity Sensitive Hashing (CMSSH) [1] models the projections from features in each modality to hash codes as binary classification problems, and learns them with boosting algorithms. Cross View Hashing (CVH) [13] extends the single-view spectral hashing to multiple views to learn hash functions via minimizing the similarity-weighted Hamming distance between hash codes of training data. Co-Regularized Hashing (CRH) [39] which is based on a boosted co-regularization framework, learns the hash functions for each bit of the hash codes by solving DC (difference of convex functions) programs, while proceeding the learning for multiple bits via a boosting procedure so that the bias introduced by the hash functions can be sequentially minimized. To tackle the high training time complexity problem, Semantic Correlation Maximization (SCM) [38] integrates semantic labels into the learning procedure and utilizes all the supervised information for training with a linear-time complexity. Unlike previous approaches that separate the optimization of the quantizer independent of maximization of domain correlation, Quantized Correlation Hashing (QCH) [32] simultaneously optimizes both processes and takes into consideration the quantization loss over domains and the underlying relation between domains. Meanwhile, the objective function of QCH is transformed to a single-modality formalization, leading to an easy optimization procedure. Semantics Preserving Hashing (SePH) [16] transforms the semantic affinities of training data into a probability distribution and approximates it with the to-be-learnt hash codes in the Hamming space via minimizing the KL-divergence. More recently, some deep hashing models have been proposed for the cross-modal search task, such as Deep Cross-Modal Hashing [10], Pairwise Relation Guided Deep Hashing [35], and Deep Visual-Semantic Hashing [2], etc. These deep model based cross-modal hashing methods have obtained competitive performance; however, our work is not a deep model, the reason is that the concentration of our work is to design the loss function. And we believe that the thought of our work can be applied in deep hashing models.

3 PROPOSED METHOD

In this section, we first define the notations used in this paper; then, show the details of our proposed method including the framework, optimization scheme and its extensions to out-of-sample data and more modalities.

3.1 Notations

For ease of representation, we assume that each sample has two modalities, e.g., image and text. However, it can be easily extended to more modalities, which is demonstrated in Section 3.5. There are n data points in the training dataset, and $\mathcal{X}^{(1)} = \{x_i^{(1)}\}_{i=1}^n \in R^{d_1}$ and $\mathcal{X}^{(2)} = \{x_i^{(2)}\}_{i=1}^n \in R^{d_2}$ denote the d_1-dimension image feature vector set and the d_2-dimension text feature vector set, respectively. Without loss of generality, we further suppose that the data points are zero-centered in both sets, i.e., $\sum_{i=1}^n x_i^{(1)} = 0$ and $\sum_{i=1}^n x_i^{(2)} = 0$. $S \in \{-1, 1\}^{n \times n}$ is the semantic similarity matrix, where $S_{ij} = 1$ if the i-th and j-th data points are semantically similar, and $S_{ij} = -1$, otherwise. $\|\cdot\|_F$ and $\|\cdot\|_1$ denote the Frobenius and L_1 norm of a vector or matrix, respectively. $sgn(\cdot)$ is an element-wise sign function which is defined as follows:

$$sgn(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0. \end{cases} \tag{1}$$

3.2 Semi-Relaxation Supervised Hashing

The goal of the supervised cross-modal hashing is to learn the k-bit binary codes for two modalities, i.e., $B = \{b_i\}_{i=1}^n \in \{-1, 1\}^{n \times k}$. At the same time, the binary codes should preserve the semantic similarity in S. To do this, we define the problem as follows:

$$\begin{aligned} &\min_B \left\| kS - BB^\top \right\|_F^2 \\ &s.t. \quad B \in \{-1, 1\}^{n \times k}. \end{aligned} \tag{2}$$

However, there are two challenging problems existing in above method: (1) It is a discrete optimization problem which is hard to solve; (2) the generation of the binary codes is independent of the features in the original space, which makes it not robust to noise. Usually, most of existing methods relax all the binary constraints in Eq. (2) to tackle the first problem [18, 38], which generate large quantization error. Some discrete methods are also proposed to tackle the first problem, which are unscalable to large-scale datasets [11, 14].

In our proposed method, we exploit a way to balance these problems. Specifically, we only relax one B in Eq. (2) by replacing it with an intermediate representation matrix T. In addition, T is consistent with the matrix of the mapping of all training samples. The objective function is defined as follows:

$$\begin{aligned} \min_{B,T,W} &\left\| kS - BT^\top \right\|_F^2 + \sum_{t=1}^2 \lambda_t \left\| T - f_t(X^{(t)}) \right\|_{2,p} \\ &+ \sum_{t=1}^2 \gamma \parallel W_t \parallel_F^2, \\ s.t. \ \ &S \in \{-1, 1\}^{n \times n}, B \in \{-1, 1\}^{n \times k}, \\ &T \in \mathbb{R}^{n \times k}, f_t(X^{(t)}) = W_t^\top \phi(X^{(t)}), \end{aligned} \tag{3}$$

where T is the intermediate representation matrix, $\lambda_t > 0$ and $\gamma > 0$ are balance parameters, and $X^{(t)}$ is the feature matrix of the t-th modality. $f_t(X^{(t)}) = W_t^\top \phi(X^{(t)})$ is the mapping function, W_t is the mapping matrix of the t-th modality, and $\phi(X)^t$ is a nonlinear embedding of $X^{(t)}$. In this paper, we use the RBF kernel mapping, i.e.:

$$\phi_i(x) = exp(\frac{- \parallel x - \hat{x}_t \parallel_2^2}{2\sigma^2}), \tag{4}$$

where $\{\hat{x}_t\}_{t=1}^m$ are the m anchor points randomly selected from the training set and σ is the kernel width, which is calculated by

$$\sigma = \frac{1}{mn} \sum_{i=1}^n \sum_{t=1}^m \parallel x_i - \hat{x}_t \parallel_2 . \tag{5}$$

By defining the optimization problem in Eq. (3), we can find the intermediate representation T can approximate the binary codes; furthermore, the semantic similarity between the intermediate representation and the binary codes can be preserved. This will significantly reduce the quantization error. In addition, the second term is used to learn the hash functions for different modalities. This means that we can learn the binary codes and hash functions simultaneously, which can further reduce the error generated by an independent quantization procedure. Note that, in the second term of Eq. (3), we use the $\ell_{2,p}$ norm instead of the Frobenius norm. The reason is that the Frobenius norm is sensitive to noise. For example, the error generated by noise will be inevitably amplified due to the squared residual. However, the $\ell_{2,p}(0 < p \leq 2)$ loss has shown the ability to alleviate sample noise [36, 37], which is defined as follows:

$$\| M \|_{2,p} = \sum_{i=1}^{n} \| m_i \|_2^p, \tag{6}$$

where $M = \{m_i\}_{i=1}^n \in \mathbb{R}^{d \times n}$. By adopting the $\ell_{2,p}$ norm, SRSH can not only suppress the influence of the potential noise, but also adapt to different levels of hash code noise.

3.3 Optimization Algorithm

In this section, we give the details of how to find a solution to the optimization problem in Eq. (3). Apparently, it is not convex because the $\ell_{2,p}(0 < p \leq 2)$ norm is used. Therefore, we propose an iterative method. First we rewrite Eq. (3) as:

$$\begin{aligned}
&\min_{B,T,W} \left\| kS - BT^{\mathsf{T}} \right\|_F^2 \\
&\quad + \sum_{t=1}^{2} \lambda_t Tr((T - W_t^{\mathsf{T}} \phi(X^{(t)})) D_t (T - W_t^{\mathsf{T}} \phi(X^{(t)}))^{\mathsf{T}}) \\
&\quad + \sum_{t=1}^{2} \gamma \| W_t \|_F^2, \\
&s.t. \quad \mathcal{S} \in \{-1,1\}^{n \times n}, \mathcal{B} \in \{-1,1\}^{n \times k}, \\
&\qquad T \in \mathbb{R}^{n \times k}, f_t(X^{(t)}) = W_t^{\mathsf{T}} \phi(X^{(t)}),
\end{aligned} \tag{7}$$

where $Tr(\cdot)$ is the trace of a matrix, D_t is a diagonal matrix with its i-th diagonal element defined as:

$$D_{ii} = \frac{1}{\frac{2}{p} \| r^i \|_2^{2-p}}, \tag{8}$$

where r^i is the i-th row of the matrix $T - W_t^{\mathsf{T}} \phi(X^{(t)})$.

Then, the iterative optimization scheme is described as follows.

Step 1: Fixing T and W_t, and updating B.

When T and W_t is fixed, we can rewrite Eq. (7) as:

$$\begin{aligned}
&\min_{B} \left\| kS - BT^{\mathsf{T}} \right\|_F^2 \\
&s.t. \quad B \in \{-1,1\}^{n \times k}.
\end{aligned} \tag{9}$$

To update B, inspired by the work [11], we first replace the Frobenius norm in Eq. (9) with the L_1 norm; therefore, the problem becomes:

$$\begin{aligned}
&\min_{B} \left\| kS - BT^{\mathsf{T}} \right\|_1 \\
&s.t. \quad B \in \{-1,1\}^{n \times k}.
\end{aligned} \tag{10}$$

Then, we have the following solution:

$$B = sgn(ST). \tag{11}$$

However, the elements of ST might be zero. To consider this, we further modify the above solution as follows:

$$B_{(i)} = resgn(ST, B_{(i-1)}), \tag{12}$$

where i is the iteration number, and $resgn(\cdot)$ is defined as:

$$resgn(var_1, var_2) = \begin{cases} 1 & var_1 > 0 \\ var_2 & var_1 = 0 \\ -1 & var_1 < 0. \end{cases} \tag{13}$$

Step 2: Fixing B and W_t, and updating T.

When B and W_t are fixed, the optimization problem can be formulated as:

$$\begin{aligned}
&\min_{T} \left\| kS - BT^{\mathsf{T}} \right\|_F^2 \\
&\quad + \sum_{t=1}^{2} \lambda_t Tr((T - W_t^{\mathsf{T}} \phi(X^{(t)})) D_t (T - W_t^{\mathsf{T}} \phi(X^{(t)}))^{\mathsf{T}}).
\end{aligned} \tag{14}$$

Setting the derivative of Eq. (14) w.r.t. T to zero, we have

$$\sum_{t=1}^{2} \lambda_t D_t T + TB^{\mathsf{T}} B - kSB - \sum_{t=1}^{2} \lambda_t D_t \phi(X^{(t)}))^{\mathsf{T}} W_t = 0, \tag{15}$$

which is a typical Sylvester equation and can be efficiently solved by using existing toolbox, such as Lyap function in Matlab.

Step 3: Fixing B and T, and updating W_t.

When B and T are fixed, the problem can be formulated as:

$$\begin{aligned}
&\min_{W_t} Tr((T - W_t^{\mathsf{T}} \phi(X^{(t)})) D_t (T - W_t^{\mathsf{T}} \phi(X^{(t)}))^{\mathsf{T}}) \\
&\quad + \sum_{t=1}^{2} \gamma \| W_t \|_F^2 .
\end{aligned} \tag{16}$$

Setting the derivative of Eq. (16) w.r.t. W_t to zero, We have :

$$W_t = (\phi(X^{(t)}) D_t \phi(X^{(t)})^{\mathsf{T}} + \gamma I)^{-1} \phi(X^{(t)}) D_t T^{\mathsf{T}}. \tag{17}$$

By repeating the above steps, we can obtain the final solution. To clearly demonstrate the proposed method, we summarize it in **Algorithm 1**.

Algorithm 1 Semi-Relaxation Supervised Hashing

```
Input: Training data matrices $X^{(1)}$, $X^{(2)}$, Similarity matrix $S$, parameters $\lambda_t, \gamma, p$,hash code length $k$, and the total iterative number $c$.
Output: Hash code matrix $B$, mapping matrix $W_t$, and intermediate representation matrix $T$.
Procedure:
1. Randomly initialize $B, T, W_t$ ;
2. Embed $\mathcal{X}^{(t)}$ into the nonlinear space with Eq. (4) and get $f_t(X^{(t)})$;
for $i = 1$ to $c$ do
    3. Fix $B$ and $T$, update $W^{(t)}$ using Eq. (17);
    4. Fix $B$ and $W^{(t)}$, update $T$ using Eq. (15);
    5. Fix $T$ and $W^{(t)}$, update $B$ using Eq. (12);
end for
return: $B, T$ and $W^{(t)}$;
```

3.4 Out-of-Sample Extension

For a new sample that is not in the training set, its binary code can be easily generated. For example, given a query sample with one of its modality $x^{(t)}$, we can obtain its hash code by using the following formula.

$$b^{(t)} = sgn(f_t(x^{(t)})) = sgn(W_t^{\mathsf{T}} \phi(x^{(t)})), \tag{18}$$

where $\phi(x^{(t)})$ is the nonlinear embedding with the RBF kernel of $x^{(t)}$ as mentioned in Section 3.2.

3.5 Extension to More Modalities

As mentioned previously, SRSH can be easily extended to more modalities. Actually, the training process for such case is nearly the same as that for bimodal case, except that the hash functions for more modalities need to be learnt independently. For example, the overall objective function for more modalities is defined as follows:

$$\begin{aligned} \min_{B,T,W} & \left\| kS - BT^{\mathsf{T}} \right\|_F^2 + \sum_{t=1}^{m} \lambda_t \left\| T - f_t(X^{(t)}) \right\|_{2,p} \\ & + \sum_{t=1}^{m} \gamma \parallel W_t \parallel_F^2, \qquad (19) \\ s.t. \ \ & B \in \{-1, 1\}^{n \times k}, T \in \mathbb{R}^{n \times k}, \\ & f_t(X^{(t)}) = W_t^T \phi(X^{(t)}), \end{aligned}$$

where m is the number of observed modalities, $X^{(t)}$ is the feature matrix of the t-th modality in training dataset. Apparently, the above problem can also be solved by the optimization algorithm proposed in Section 3.3.

4 EXPERIMENTS

To test the performance of our proposed method, we carried out extensive experiments on three widely-used benchmark datasets, i.e., Wiki [23], MIRFlickr-25K [8], and NUS-WIDE [3]. All of the datasets are with two modalities, i.e., image and text. We also compared it with eight state-of-the-art cross-modal hashing methods.

4.1 Datasets

Wiki: It is collected from the Wikipedia with 2,866 image-text pairs. Each instance is annotated with one of 10 semantic classes. In addition, the visual modality of each instance is represented by a 128-dimension bag-of-visual $SIFT$ feature vector, and the textual one is represented by a 10-dimension topic vector. On this dataset, we use 75% of the dataset as the training set, the rest 25% as the query set.

MIRFlickr-25K: It consists of 25,000 instances collected from Flickr, each being an image annotated by some textual tags from 24 unique labels. The visual content is described by a 150-dimension edge histogram and the textual content is represented as a 500-dimension feature vector derived from PCA on its binary tagging vector w.r.t the remaining textual tags. We randomly select 75% instances as the training set; the remaining 25% instances are used as the query set.

NUS-WIDE: The NUS-WIDE dataset is a real-world web image dataset collected by the Lab for Media Search in National University of Singapore. It contains 269,648 images crawled from Flickr, together with its associated textual tags. Each instance is manually annotated with at least one of 81 provided labels. Considering some labels are scarce, we select 10 most common concepts and the corresponding 186,577 images as the final dataset. Each image-text pair is annotated by at least 1 of 10 concepts. For each instance, the visual view is represented by a 500-dimension bag-of-visual SIFT feature vector and the textual view is represented by a 1,000-dimension vector. We randomly select 1% of the dataset as the query set and the rest as the training set.

Considering the computational cost, for all methods on MIRFlickr-25K and NUS-WIDE, 5,000 and 10,000 samples are randomly selected from the original training set to train the proposed and all baselines, respectively.

4.2 Baselines and Evaluation Metric

We compare our proposed SRSH with eight sate-of-the-art hashing methods for cross-modal search task, i.e., IMH[25], CVH[13], SCM-orth[38], SCM-seq[38], LSSH[40], CMFH[5], SePH-km[16], and CCQ[19]. They can be divided into two categories: IMH, LSSH, CMFH and CCQ are unsupervised ones, while CVH, SCM-orth, SCM-seq, and SePH-km are supervised ones. Source codes of most baselines are kindly provided by the authors. We carefully tune the parameters of these models and report their best results. The parameters of SRSH are also selected by a validation procedure, i.e., $\lambda_1 = 0.7, \lambda_2 = 0.3, p = 1.2, \gamma = 0.05$. In addition, the iteration number c is to 4.

The performance of all methods is evaluated by the widely-used Mean Average Precision (MAP). For a query q, the average precision (AP) is defined as:

$$AP(q) = \frac{1}{L_q} \sum_{r=1}^{n} P_q(r) \delta_q(r), \tag{20}$$

where L_q is the number of ground-truth neighbors of query q in the database, n is the number of entities in the database, $P_q(r)$ denotes the precision of the top r retrieved entities, and $\delta_q(r) = 1$ if the r-th retrieved entity is a ground-truth neighbour and $\delta_q(r) = 0$, otherwise. The ground-truth neighbors are defined as those sharing at least one semantic label. The MAP is defined as:

$$MAP = \frac{1}{|Q|} \sum_{i=1}^{|Q|} AP(q_i), \tag{21}$$

where $|Q|$ is the size of the query set Q.

We also plot the precision-recall and top-N precision curves on some cases.

4.3 Results and Discussions

4.3.1 Results on Wiki. The MAP values of SRSH and all of the baselines on Wiki are summarized in Table 1, including the results of the "Image-to-Text" and "Text-to-Image" search tasks. From Table 1, we can observe that

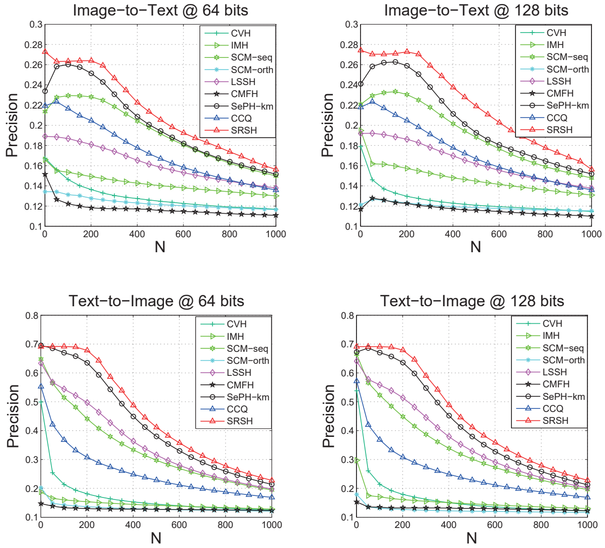

Figure 1: Top-N precision curves on Wiki

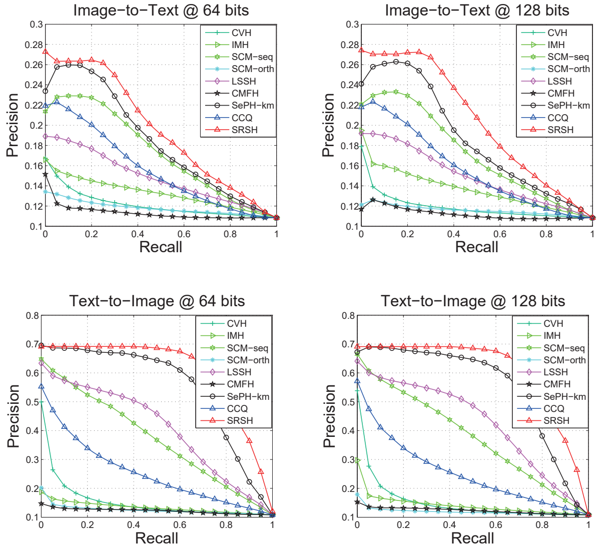

Figure 2: Precision-recall curves on Wiki

Table 1: Performance (MAP) comparison on Wiki ($I \rightarrow T$ means the search task of Image-to-Text, and vice versa). The best results are shown in boldface.

Task	Method	16 bits	24 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CVH	0.1435	0.1383	0.1368	0.1321	0.1272
	IMH	0.1667	0.1683	0.1655	0.1702	0.1798
	SCM-orth	0.1656	0.1514	0.1479	0.1460	0.1409
	SCM-seq	0.2629	0.2665	0.2646	0.2804	0.2821
	LSSH	0.1940	0.2001	0.1962	0.2105	0.2107
	CMFH	0.1228	0.1257	0.1241	0.1251	0.1243
	CCQ	0.2039	0.2063	0.2078	0.2045	0.2090
	SePH-km	0.2796	0.2822	0.2820	0.3076	0.3136
	SRSH	**0.3026**	**0.3186**	**0.3609**	**0.3642**	**0.3812**
$T \rightarrow I$	CVH	0.1579	0.1551	0.1568	0.1579	0.1567
	IMH	0.1360	0.1403	0.1385	0.1401	0.1442
	SCM-orth	0.1671	0.1515	0.1411	0.1281	0.1218
	SCM-seq	0.3779	0.3792	0.3917	0.4223	0.4300
	LSSH	0.4112	0.4339	0.4461	0.4756	0.4963
	CMFH	0.1294	0.1330	0.1348	0.1339	0.1332
	CCQ	0.2649	0.2718	0.2769	0.2660	0.2779
	SePH-km	0.6378	0.6390	0.6451	0.6661	0.6705
	SRSH	**0.6545**	**0.6990**	**0.7372**	**0.7585**	**0.7569**

Table 2: Performance (MAP) Comparison on MIRFlickr-25K. The best results are shown in boldface.

Task	Method	16 bits	24 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	CVH	0.5719	0.5701	0.5698	0.5676	0.5665
	IMH	0.5635	0.5660	0.5661	0.5673	0.5681
	SCM-orth	0.5955	0.5882	0.5845	0.5772	0.5716
	SCM-seq	0.6443	0.6531	0.6582	0.6630	0.6678
	LSSH	0.5562	0.5670	0.5624	0.5647	0.5663
	CMFH	0.5667	0.5694	0.5704	0.5689	0.5698
	CCQ	0.5668	0.5667	0.5665	0.5671	0.5670
	SePH-km	0.6843	0.6860	0.6873	0.6882	0.6874
	SRSH	**0.7071**	**0.6927**	**0.6946**	**0.7105**	**0.7128**
$T \rightarrow I$	CVH	0.5742	0.5730	0.5715	0.5704	0.5692
	IMH	0.5627	0.5642	0.5646	0.5652	0.5691
	SCM-orth	0.6023	0.5935	0.5883	0.5778	0.5694
	SCM-seq	0.6479	0.6611	0.6664	0.6743	0.6805
	LSSH	0.5619	0.5565	0.5663	0.5719	0.5766
	CMFH	0.5682	0.5706	0.5716	0.5705	0.5708
	CCQ	0.5730	0.5737	0.5736	0.5740	0.5740
	SePH-km	0.7389	0.7409	0.7456	0.7476	0.7497
	SRSH	**0.7799**	**0.7813**	**0.7866**	**0.8007**	**0.8020**

- SRSH outperforms all of the baselines in all cases, which well demonstrates its effectiveness.
- Generally, all of the methods are doing better at Text-to-Image than Image-to-Text task. The main reason is that texts can better describe the content of an image-text pair than the image.
- With the hash code length increasing, the performance of SRSH generally keeps increasing, which means that utilizing longer hash codes can better preserve semantic similarity.

To gain deep insights into SRSH and all baselines, we further plot the top-N precision and precision-recall curves of the cases with 64 and 128 bits, which are shown in Figure 1 & 2. From these figures, we can observe similar results to those in Table 1. For example, SRSH consistently outperforms all other methods. In addition, from Figure 1 & 2 and Table 1, we can observe that those methods with orthogonality constraints, e.g., SCM-orth and CVH, perform badly on most cases with the code length increasing. A reasonable explanation is that these orthogonality constraints sometimes lead to additional problems. For example, their first few projects may have high variance and the corresponding bits are discriminative. However, with the code length increasing, the binary codes may be dominated by low variance bits

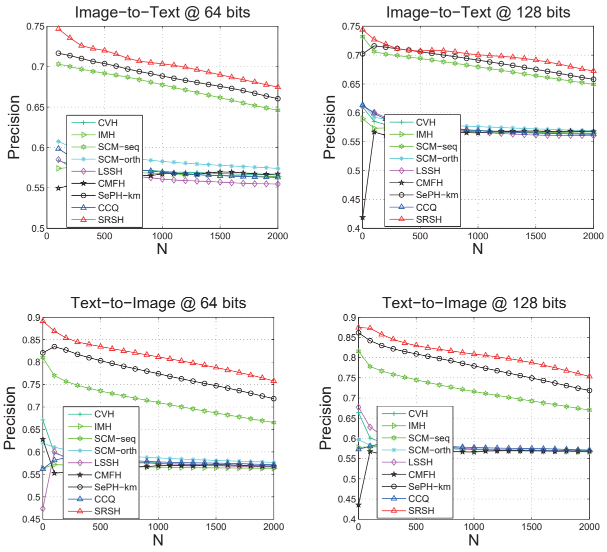

Figure 3: Top-N precision curves on MIRFlickr-25k

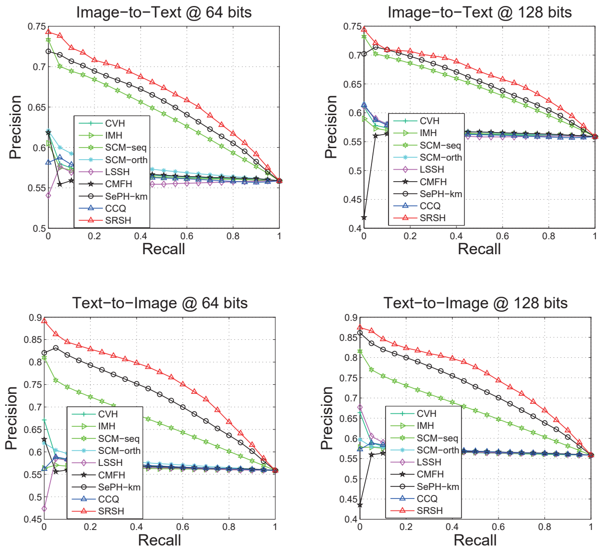

Figure 4: Precision-recall curves on MIRFlickr-25k

Table 3: Performance (MAP) comparison on NUS-WIDE. The best results are shown in boldface

Task	Method	16 bits	24 bits	32 bits	64 bits	128 bits
$I \to T$	CVH	0.3752	0.3691	0.3652	0.3570	0.3514
	IMH	0.3567	0.3585	0.3574	0.3640	0.3607
	SCM-orth	0.3910	0.3813	0.3763	0.3648	0.3579
	SCM-seq	0.5148	0.5183	0.5293	0.5336	0.5315
	LSSH	0.3516	0.3629	0.3643	0.3613	0.3757
	CMFH	0.3577	0.3547	0.3563	0.3549	0.3564
	CCQ	0.3420	0.3430	0.3432	0.3441	0.3436
	SePH-km	0.5369	0.5353	0.5440	0.5449	0.5510
	SRSH	**0.5903**	**0.5688**	**0.5857**	**0.6174**	**0.6331**
$T \to I$	CVH	0.3744	0.3697	0.3662	0.3586	0.3531
	IMH	0.3473	0.3498	0.3479	0.3547	0.3529
	SCM-orth	0.3868	0.3738	0.3680	0.3573	0.3501
	SCM-seq	0.4982	0.5036	0.5118	0.5203	0.5197
	LSSH	0.3507	0.3634	0.3590	0.3687	0.3801
	CMFH	0.3612	0.3575	0.3601	0.3577	0.3595
	CCQ	0.3644	0.3648	0.3655	0.3659	0.3657
	SePH-km	0.6203	0.6242	0.6358	0.6405	0.6391
	SRSH	**0.6627**	**0.6547**	**0.6719**	**0.7323**	**0.7397**

4.3.2 Results on MIRFlickr-25K. The results on MIRFlickr-25K are displayed in Table 2, and Figure 3 & 4. The MAP values on Image-to-Text and Text-to-Image tasks are listed in Table 2; the top-N precision and precision-recall curves are plotted in Figure 3 & 4, respectively. From these results, we have the following observations:

- SRSH outperforms all baselines in all cases.
- Similar to that on Wiki, SRSH, SePH-km and SCM-seq are doing better than other methods.
- Most methods are doing better at Text-to-Image than Image-to-Text, which is consistent with that on Wiki. This further confirms the fact that text can better describe the topic of the image-text pair than image.

4.3.3 Results on NUS-WIDE. The MAP values of all methods on NUS-WIDE are listed in Table 3; the top-N precision and precision-recall curves of the cases with 64 and 128 bits are plotted in Figure 5 & 6, respectively. From these results, we have the following observations, which are very similar to those on Wiki and MIRFlickr-25K:

- SRSH outperforms all baselines in all cases.
- Especially, from Figure 5 & 6, we can observe that SRSH is doing much better than other methods at the beginning, e.g., N is small. This means that SRSH returns highly related samples when N is small, which is very important in retrieval task.
- SRSH, SePH-km and SCM-seq are doing much better than other methods.
- Generally, the results of all methods on Text-to-Image task are better than those on Image-to-Text task, which is consistent with that on Wiki and MIRFlickr-25K.

To summarize, from the results on Wiki, MIRFlickr-25K and NUS-WIDE, we can conclude that the proposed SRSH can work well on these datasets, and outperform other state-of-the-art cross-modal hashing methods, which confirms its effectiveness.

4.3.4 Parameter Sensitivity Analysis. The parameters of SRSH may have potential influence on the performance. Especially, the parameter p controls the robustness of SRSH. To confirm this, we conduct experiments on Wiki to analyze its influence on the performance. In Figure 7, we plot the MAP curves of the cases with 32 bits of both the Image-to-Text and Text-to-Image tasks by varying p from 0.2 to 1.8 with step size 0.2. From this figure, we can observe that SRSH is indeed influenced by p. In general, with p increasing

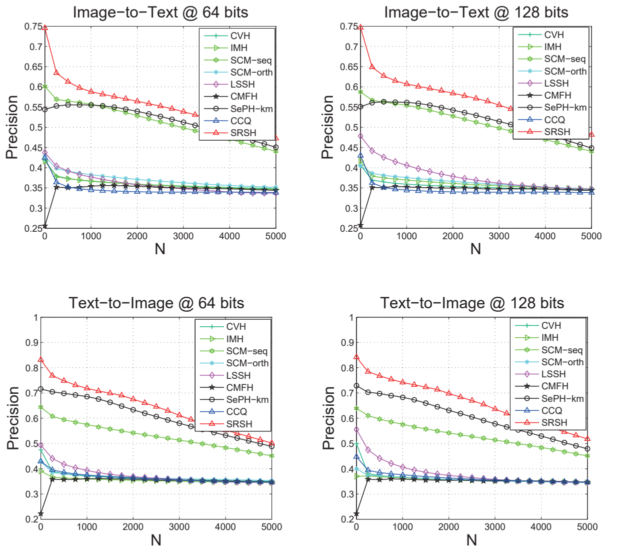

Figure 5: Top-N precision curves on NUS-WIDE

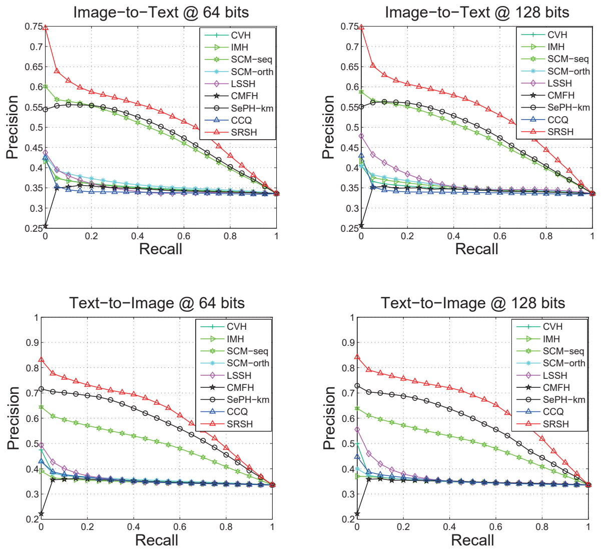

Figure 7: Precision-recall curves on NUS-WIDE

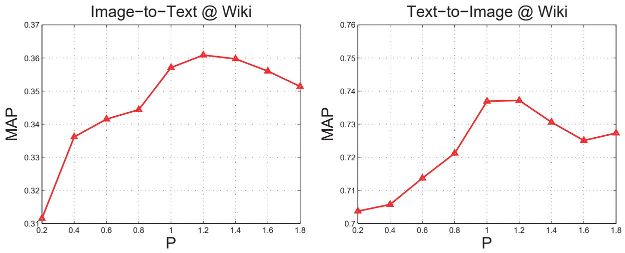

Figure 6: Sensitivity analysis of parameter p on Wiki with 32 bits

Table 4: Training time comparison on MIRFlickr-25k (in seconds).

Method	16 bits	24 bits	32 bits	64 bits	128 bits
CVH	3.2	3.0	3.1	3.2	3.3
IMH	28.8	38.3	42.7	38.3	41.7
SCM-orth	2.2	1.9	2.5	2.0	2.5
SCM-seq	37.5	53.7	87.6	140.3	271.5
LSSH	117.6	155.0	141.6	163.6	162.9
CMFH	0.6	0.6	0.6	0.7	1.0
CCQ	7.5	23.5	23.9	102.1	520.6
SePH-km	581.7	717.9	689.6	1030.2	1840.9
SRSH	12.1	16.3	14.5	21.3	28.9

($p < 1.2$), the performance of SRSH becomes better; however, when $p > 1.2$, the performance degrades quickly.

4.4 Time Cost Analysis

To demonstrate the efficiency of SRSH, we further compare the training time of all methods on MIRFlickr-25K, the result is summarized in Table 4. The length of hash code varies from 16 to 128. From this table, we can observe that the training time of SRSH is acceptable. Especially, it uses much less training time than SCM-seq, CCQ, LSSH and SePH-km on cases with long code length. Note that SCM-orth and CMFH do not use the similarity matrix directly; therefore, they are much faster than other methods.

5 CONCLUSION

In this paper, we present a novel supervised cross-modal hashing method, i.e., Semi-Relaxation Supervised Hashing (SRSH). Given a similarity matrix, it mainly focuses on tackling three problems: (1) How to reduce the quantization error; (2) how to solve the discrete optimization problem; (3) how to learn the binary codes and hash functions simultaneously. To tackle these problems, it relaxes a part of the binary constraints and replaces one binary matrix with an intermediate representation matrix. An iterative algorithm is proposed to solve the optimization problem. Extensive experimental results on Wiki, MIRFlickr-25k and NUS-WIDE demonstrate that SRSH outperforms several state-of-the-art baselines for cross-modal search task.

6 ACKNOWLEDGEMENTS

This work was partially supported by National Natural Science Foundation of China (61173068, 61573212), Program for New Century Excellent Talents in University of the Ministry of Education, Key Research and Development Program of Shandong Province (2016GGX101044).

REFERENCES

[1] Michael M Bronstein, Alexander M Bronstein, Fabrice Michel, and Nikos Paragios. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3594–3601.

[2] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. 2016. Deep Visual-Semantic Hashing for Cross-Modal Retrieval. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*. 1445–1454.

[3] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 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*. 48.

[4] Qi Dai, Jianguo Li, Jingdong Wang, and Yu-Gang Jiang. 2016. Binary optimized hashing. In *Proceedings of ACM International Conference on Multimedia*. 1247–1256.

[5] Guiguang Ding, Yuchen Guo, and Jile Zhou. 2014. Collective matrix factorization hashing for multimodal data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2075–2082.

[6] Aristides Gionis, Piotr Indyk, Rajeev Motwani, and others. 1999. Similarity search in high dimensions via hashing. In *Proceedings of International Conference on Very Large Data Bases*. 518–529.

[7] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 35 (2013), 2916–2929.

[8] Mark J Huiskes and Michael S Lew. 2008. The MIR flickr retrieval evaluation. In *Proceedings of ACM International Conference on Multimedia Information Retrieval*. 39–43.

[9] Qing-Yuan Jiang and Wu-Jun Li. 2015. Scalable graph hashing with feature transformation.. In *Proceedings of International Joint Conference on Artificial Intelligence*. 2248–2254.

[10] Qing-Yuan Jiang and Wu-Jun Li. 2016. Deep Cross-Modal Hashing. *arXiv preprint arXiv:1602.02255* (2016).

[11] Wang-Cheng Kang, Wu-Jun Li, and Zhi-Hua Zhou. 2016. Column sampling based discrete supervised hashing.. In *Proceedings of AAAI Conference on Artificial Intelligence*. 1230–1236.

[12] Weihao Kong and Wu-Jun Li. 2012. Isotropic hashing. In *Proceedings of Advances in Neural Information Processing Systems*. 1646–1654.

[13] Shaishav Kumar and Raghavendra Udupa. 2011. Learning hash functions for cross-view similarity search. In *Proceedings of International Joint Conference on Artificial Intelligence*. 1360.

[14] Guosheng Lin, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, and David Suter. 2014. Fast supervised hashing with decision trees for high-dimensional data. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1963–1970.

[15] Guosheng Lin, Chunhua Shen, David Suter, and Anton Van Den Hengel. 2013. A general two-step approach to learning-based hashing. In *Proceedings of IEEE International Conference on Computer Vision*. 2552–2559.

[16] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. 2015. Semantics-preserving hashing for cross-view retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3864–3872.

[17] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang. 2014. Discrete graph hashing. In *Proceedings of Advances in Neural Information Processing Systems*. 3419–3427.

[18] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. 2012. Supervised hashing with kernels. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2074–2081.

[19] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S Yu. 2016. Composite correlation quantization for efficient multimodal retrieval. In *Proceedings of ACM International Conference on Research and Development in Information Retrieval*. 579–588.

[20] Liqiang Nie, Meng Wang, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Oracle in Image Search: A Content-Based Approach to Performance Prediction. *ACM Transactions on Information System* 30, 2 (2012), 13:1–13:23.

[21] Liqiang Nie, Shuicheng Yan, Meng Wang, Richang Hong, and Tat-Seng Chua. 2012. Harvesting Visual Concepts for Image Search with Complex Queries. In *Proceedings of ACM International Conference on Multimedia*. 59–68.

[22] Mohammad Norouzi and David M Blei. 2011. Minimal loss hashing for compact binary codes. In *Proceedings of International Conference on Machine Learning*. 353–360.

[23] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of ACM International Conference on Multimedia*. 251–260.

[24] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. 2015. Supervised discrete hashing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 37–45.

[25] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *Proceedings of ACM International Conference on Management of Data*. 785–796.

[26] Christoph Strecha, Alex Bronstein, Michael Bronstein, and Pascal Fua. 2012. LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 1 (2012), 66–78.

[27] Jinhui Tang, Zechao Li, Meng Wang, and Ruizhen Zhao. 2015. Neighborhood Discriminant Hashing for Large-Scale Image Retrieval. *IEEE Transactions on Image Processing* 24, 9 (2015), 2827–2840.

[28] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2010. Sequential projection learning for hashing with compact codes. In *Proceedings of International Conference on Machine Learning*. 1127–1134.

[29] Jian Wang, Xin-Shun Xu, Shanqing Guo, Lizhen Cui, and Xiao-Lin Wang. 2016. Linear unsupervised hashing for ANN search in Euclidean space. *Neurocomputing* 171 (2016), 283–292.

[30] Min Wang, Wengang Zhou, Qi Tian, Zhengjun Zha, and Houqiang Li. 2016. Linear Distance Preserving Pseudo-Supervised and Unsupervised Hashing. In *Proceedings of ACM International Conference on Multimedia*. 1257–1266.

[31] Yair Weiss, Antonio Torralba, and Rob Fergus. 2009. Spectral hashing. In *Proceedings of Advances in Neural Information Processing Systems*. 1753–1760.

[32] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. 2015. Quantized correlation hashing for fast cross-modal search.. In *Proceedings of International Joint Conference on Artificial Intelligence*. 3946–3952.

[33] Xin-Shun Xu. 2016. Dictionary learning based hashing for cross-modal retrieval. In *Proceedings of ACM International Conference on Multimedia*. 177–181.

[34] Ting-Kun Yan, Xin-Shun Xu, Shanqing Guo, Zi Huang, and Xiao-Lin Wang. 2016. Supervised robust discrete multimodal hashing for cross-media retrieval. In *Proceedings of ACM International on Conference on Information and Knowledge Management*. 1271–1280.

[35] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. 2017. Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval. In *Proceedings of AAAI Conference on Artificial Intelligence*. 1618–1625.

[36] Yang Yang, Zhigang Ma, Yi Yang, Feiping Nie, and Heng Tao Shen. 2015. Multitask spectral clustering by exploring intertask correlation. *IEEE Transactions on Cybernetics* 45, 5 (2015), 1083–1094.

[37] Yang Yang, Zheng-Jun Zha, Yue Gao, Xiaofeng Zhu, and Tat-Seng Chua. 2014. Exploiting web images for semantic video indexing via robust sample-specific loss. *IEEE Transactions on Multimedia* 16, 6 (2014), 1677–1689.

[38] Dongqing Zhang and Wu-Jun Li. 2014. Large-scale supervised multimodal hashing with semantic correlation maximization.. In *Proceedings of AAAI Conference on Artificial Intelligence*. 7.

[39] Yi Zhen and Dit-Yan Yeung. 2012. Co-regularized hashing for multimodal data. In *Proceedings of Advances in Neural Information Processing Systems*. 1376–1384.

[40] Jile Zhou, Guiguang Ding, and Yuchen Guo. 2014. Latent semantic sparse hashing for cross-modal similarity search. In *Proceedings of ACM International Conference on Research and Development in Information Retrieval*. 415–424.

[41] Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. 2013. Linear cross-modal hashing for efficient multimedia search. In *Proceedings of ACM International Conference on Multimedia*. 143–152.