Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization

Dongqing Zhang[†] and **Wu-Jun Li**[‡] *
[†] Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[‡] National Key Laboratory for Novel Software Technology
Department of Computer Science and Technology, Nanjing University, China
zdq@sjtu.edu.cn, liwujun@nju.edu.cn

Abstract

Due to its low storage cost and fast query speed, hashing has been widely adopted for similarity search in multimedia data. In particular, more and more attentions have been payed to multimodal hashing for search in multimedia data with multiple modalities, such as images with tags. Typically, supervised information of semantic labels is also available for the data points in many real applications. Hence, many supervised multimodal hashing (SMH) methods have been proposed to utilize such semantic labels to further improve the search accuracy. However, the training time complexity of most existing SMH methods is too high, which makes them unscalable to large-scale datasets. In this paper, a novel SMH method, called semantic correlation maximization (SCM), is proposed to seamlessly integrate semantic labels into the hashing learning procedure for large-scale data modeling. Experimental results on two real-world datasets show that SCM can significantly outperform the state-of-the-art SMH methods, in terms of both accuracy and scalability.

Introduction

Recent years have witnessed the great success of hashing techniques for scalable similarity search in many real applications (Torralba, Fergus, and Weiss 2008; Salakhutdinov and Hinton 2009; Kong and Li 2012a; He et al. 2012; Tseng et al. 2012; Dean et al. 2013; Wang et al. 2013; Xu et al. 2013; Zhang et al. 2014). The basic idea of hashing is to transform the data points from the original feature space into a Hamming space with binary hash codes, where the storage cost can be substantially reduced and the query speed can be dramatically improved. Although a lot of hashing methods have been proposed, most of them focus on data in a single-view space. That is to say, most existing hashing methods are unimodal (Weiss, Torralba, and Fergus 2008; Andoni and Indyk 2008; Kulis and Darrell 2009; Lin, Ross, and Yagnik 2010; Wang, Kumar, and Chang 2010; Norouzi and Fleet 2011; Kong, Li, and Guo 2012; Heo et al. 2012; Liu et al. 2012a; Kong and Li 2012b; Norouzi, Fleet, and Salakhutdinov 2012; Strecha et al. 2012; Ge et al. 2013; Neyshabur et al. 2013).

*Wu-Jun Li is the corresponding author.

As the development of information technology, more and more multimodal data have been available in many applications, especially in multimedia domains (Barnard and Forsyth 2001; Chua et al. 2009; Song et al. 2011; Chen et al. 2012; Wu et al. 2014). For example, a Flickr image may have tags associated with it, and a web image can have surrounding texts relevant to it. How to leverage such multimodal data to conduct cross-view similarity search (Rasiwasia et al. 2010; Hwang and Grauman 2012; Sharma et al. 2012; Gong et al. 2014) has become a challenging but interesting research problem. Hence, there also exist several multimodal hashing methods for fast similarity search in multimodal data (Bronstein et al. 2010; Kumar and Udupa 2011; Song et al. 2011; Gong and Lazebnik 2011; Zhang, Wang, and Si 2011; Liu et al. 2012b; Zhen and Yeung 2012a; 2012b; Song et al. 2013; Zhai et al. 2013; Ou et al. 2013; Rastegari et al. 2013).

Existing multimodal hashing methods can be divided into two main categories: *multi-source hashing* (MSH) and *cross-modal hashing* (CMH). MSH is also called multiple feature hashing (Song et al. 2011) or composite hashing (Zhang, Wang, and Si 2011), which aims at learning better codes by leveraging auxiliary views than unimodal hashing. MSH assumes that all the views should be provided for a query point when performing search, which are typically not feasible for many multimedia applications.

The CMH methods are much more popular because only one view is needed for a query point in CMH. For example, all the tasks of image-to-image, text-to-image, and image-to-text retrieval can be performed with CMH in the Flickr image datasets. According to whether *supervised information* is used or not, CMH methods can be further divided into two subcategories: *unsupervised CMH* and *supervised CMH*. Unsupervised CMH methods, such as the method in (Gong and Lazebnik 2011), mostly rely on canonical correlation analysis (CCA) (Hotelling 1936), which maps two views, such as visual and textual views, into a common latent space where the correlation between the two views is maximized. This space is cross-modal, which means that entities in either visual or textual view can be transformed into this space, and thus image-to-image, text-to-image, and image-to-text retrieval tasks can all be handled in the same manner.

In many real applications, besides the multimodal (multiview) feature information, supervised information like *se-*

mantic labels is also available for the data points. Such supervised information, typically provided by people, is very discriminative for hashing function learning. Hence, supervised CMH methods, which can utilize supervised information for hashing, have attracted more and more attention from researchers. Representative supervised CMH methods include CMSSH (Bronstein et al. 2010), *cross view hashing* (CVH) (Kumar and Udupa 2011), *multimodal latent binary embedding* (MLBE) (Zhen and Yeung 2012b), and *co-regularized hashing* (CRH) (Zhen and Yeung 2012a). CMSSH (Bronstein et al. 2010) aims at learning two hash functions for two modalities using eigen-decomposition and boosting. CVH (Kumar and Udupa 2011) extends spectral hashing (Weiss, Torralba, and Fergus 2008) to the multimodal setting. MLBE (Zhen and Yeung 2012b) directly learns the binary hash codes with latent variable models. CRH (Zhen and Yeung 2012a) is learned by solving the difference of convex function programs, while the learning for multiple bits is performed by a boosting procedure.

Although *supervised multimodal hashing* (SMH) methods, mainly including the supervised CMH methods mentioned above, have achieved promising results in many real applications, the training time complexity of most existing SMH methods is too high. More specifically, the training time complexity of these methods is at least $O(N_{S^{xy}})$, where $N_{S^{xy}}$ is the number of observed similar or dissimilar pairs between the two views x and y. Assuming that there are n data points with semantic labels in the training set, $N_{S^{xy}}$ can be as large as n^2. Hence, existing SMH methods cannot be scalable to large-scale datasets. To handle large-scale datasets, most of them have to sample only a small subset of the whole training set or sample only a subset of the similar or dissimilar pairs from the training set. Both of these two sampling strategies will deteriorate the accuracy because the supervised information cannot be fully utilized.

In this paper, a novel SMH method, called *semantic correlation maximization* (SCM), is proposed to seamlessly integrate semantic labels into the hashing learning procedure for large-scale data modeling. The main contributions of our SCM hashing method are summarized as follows:

- By avoiding explicitly computing the pairwise similarity matrix, our SCM method can utilize all the supervised information for training with linear-time complexity, which is much more scalable than existing SMH methods.
- A sequential learning method is proposed to learn the hash functions bit by bit. The solution of the hash function for each bit has a closed-form solution. Hence, no hyper-parameters and stopping conditions are needed for tuning in SCM.
- Experimental results on two real-world datasets show that SCM can significantly outperform the state-of-the-art SMH methods, in terms of both accuracy and scalability.

Please note that we only focus on the cross-modal hashing in this paper because cross-modal hashing is much more popular than multi-source hashing in real applications. Moreover, it is easy to adapt the proposed method for multi-source hashing problems.

Notation and Problem Definition

For ease of presentation, here we describe the SMH problem with only two modalities, which can be easily extended to the cases with more than two modalities.

Let n denote the number of training entities (data points), x and y denote the two modalities (views) for each entity. We use $\{x_1, x_2, ..., x_n | x_i \in \mathbb{R}^{d_x}\}$ and $\{y_1, y_2, ..., y_n | y_i \in \mathbb{R}^{d_y}\}$ to denote the feature vectors of the two modalities in the original space, where d_x and d_y are the dimensions of feature space in each modality. These feature vectors form the rows of the data matrix $X \in \mathbb{R}^{n \times d_x}$ and $Y \in \mathbb{R}^{n \times d_y}$, respectively. For example, in the Flickr image search application, x_i is the image content information of the entity i, and y_i is the tag information of the entity i. Without loss of generality, we assume that the data points are zero-centered, i.e., $\sum_{i=1}^{n} x_i = \mathbf{0}$ and $\sum_{i=1}^{n} y_i = \mathbf{0}$. Furthermore, we assume that both modalities are observed (available) for all the data points in the *training set*. But our model can be easily extended to the cases with missing modality for some training points. Note that for a query point, only one modality is needed for search. That is to say, given a query entity q, we can perform search when either x_q or y_q is available. This is the key difference between the cross-modal hashing in this paper and the multi-source hashing in (Song et al. 2011) and (Zhang, Wang, and Si 2011).

Besides the feature vectors of the two modalities x and y, we also have *semantic labels* for each training entity in SMH. These labels are denoted by the label vectors $\{l_1, l_2, ..., l_n | l_i \in \{0,1\}^m\}$, where m is the total number of categories. Here, we assume that each entity belongs to at least one of the m categories. $l_{i,k} = 1$ denotes that the ith entity belongs to the kth semantic category. Otherwise, $l_{i,k} = 0$.

The goal of SMH is to learn two hashing functions for the two modalities: $f(x) : \mathbb{R}^{d_x} \to \{-1,1\}^c$ and $g(y) : \mathbb{R}^{d_y} \to \{-1,1\}^c$, where c is the length of the binary hash code. These two hashing functions map the feature vectors in the corresponding modality into a common hamming space which should preserve the *semantic similarity* of the labels. Although many different kinds of functions can be used to define $f(x)$ and $g(y)$, we adopt the commonly used hashing function form, which is defined as follows:

$$f(x) = sgn(W_x^T x),$$
$$g(y) = sgn(W_y^T y),$$

where $sgn(\cdot)$ denotes the element-wise sign function, $W_x = [w_x^{(1)}, w_x^{(2)}, ..., w_x^{(c)}] \in \mathbb{R}^{d_x \times c}$ and $W_y = [w_y^{(1)}, w_y^{(2)}, ..., w_y^{(c)}] \in \mathbb{R}^{d_y \times c}$ are the projection matrices.

Hence, the problem of our SMH is to learn the two projection matrices W_x and W_y from the label vectors $\{l_1, l_2, ..., l_n\}$ and the feature matrices X and Y.

Our Methodology

In this section, we describe the details of the model and the learning algorithm of our SCM hashing method.

Model Formulation

To leverage semantic labels for SMH, we construct the pairwise semantic similarity by the cosine similarity between the semantic label vectors. More specifically, the similarity between the ith entity and the jth entity is defined as follows:

$$\tilde{S}_{ij} = \frac{l_i \cdot l_j}{\|l_i\|_2 \, \|l_j\|_2}, \tag{1}$$

where $l_i \cdot l_j$ denotes the inner product between the two label vectors l_i and l_j, and $\|l_i\|_2$ denotes the two-norm (length) of the label vector l_i.

We use a $n \times m$ matrix $\tilde{L}$ to store the label information, with $\tilde{L}_{ik} = \frac{l_{i,k}}{\|l_i\|_2}$. Here, $\tilde{L}_{ik}$ denotes the element at the ith row and the kth column in the matrix $\tilde{L}$. Then, we can write the similarity matrix as $\tilde{S} = \tilde{L}\tilde{L}^T$, where the element at the ith row and the jth column in $\tilde{S}$ is $\tilde{S}_{ij}$. We perform element-wise linear transformation on $\tilde{S}$ to get our final *semantic similarity matrix* $S \in [-1, 1]^{n \times n}$:

$$S = 2\tilde{S} - E = 2\tilde{L}\tilde{L}^T - \mathbf{1}_n\mathbf{1}_n^T, \tag{2}$$

where $\mathbf{1}_n$ is an all-one column vector with length n, and E is an all-one matrix. Note that we use the semantic similarity matrix S of size $n \times n$ here just for ease of understanding. In the following learning algorithm, this matrix will not be *explicitly* computed. This is one of the key contributions of our method to avoid the high time complexity.

Because our focus is on cross-modal similarity search, the two hashing functions should preserve the semantic similarity cross modalities. More specifically, we try to reconstruct the *semantic similarity matrix* by the learned hash codes. Hence, the objective function of our model is to minimize the following squared error:

$$\min_{f,g} \sum_{i,j} \left(\frac{1}{c} f(x_i)^T g(y_j) - S_{ij} \right)^2. \tag{3}$$

In matrix form, we can rewrite the problem in (3) as follows:

$$\min_{W_x, W_y} \left\| sgn(XW_x) sgn(YW_y)^T - cS \right\|_F^2 \tag{4}$$

This objective function is simpler than existing methods and offers a clearer connection between the learned hash codes and the semantic similarity. However, it is NP hard to directly compute the best binary functions of the problem in (4). In the next subsections, we'll discuss our algorithms which can efficiently learn the binary functions.

Learning for Orthogonal Projection

One common way to approximately solve the NP hard problem in (4) is to apply spectral relaxation (Weiss, Torralba, and Fergus 2008) and impose orthogonality constraints in order to make the bits between different hashing functions balanced and uncorrelated, which can be formulated as follows:

$$\begin{aligned} \max_{W_x, W_y} \; & \left\| (XW_x)(YW_y)^T - cS \right\|_F^2 \\ \text{s.t.} \; & W_x^T X^T X W_x = nI_c \\ & W_y^T Y^T Y W_y = nI_c, \end{aligned} \tag{5}$$

where I_c denotes an identity matrix of size $c \times c$.

With simple algebra, we can transform the objective function in (5) into the following form:

$$\begin{aligned} & \left\| (XW_x)(YW_y)^T - cS \right\|_F^2 \\ =& tr[((XW_x)(YW_y)^T - cS)((XW_x)(YW_y)^T - cS)^T] \\ =& tr[(XW_x)(YW_y)^T(YW_y)(XW_x)^T] \\ & \quad - 2c \cdot tr[(XW_x)^T S (YW_y)] + tr(c^2 S^T S) \\ =& -2c \cdot tr(W_x^T X^T S Y W_y) + cn^2 + tr(c^2 S^T S) \\ =& -2c \cdot tr(W_x^T X^T S Y W_y) + const, \end{aligned}$$

where $tr()$ denotes the trace of a matrix, and $const$ is a constant independent of the variables W_x and W_y.

Then, we can reformulate the problem in (5) as the following equivalent quadratically constrained quadratic program:

$$\begin{aligned} \max_{W_x, W_y} \; & tr(W_x^T X^T S Y W_y) \\ \text{s.t.} \; & W_x^T X^T X W_x = nI_c \\ & W_y^T Y^T Y W_y = nI_c. \end{aligned} \tag{6}$$

In (6), the term $X^T S Y$ actually measures the correlation between the two modalities with respect to the *semantic labels*. This correlation is called *semantic correlation* in this paper because the semantic labels are seamlessly integrated into the correlation computation. From (6), we can find that the goal of our method is to maximize the semantic correlation. Hence, we name our method as *semantic correlation maximization* (SCM).

It is very interesting to find that our SCM method will degenerate to the CCA formulation when $S = I_n$.

We can prove that the problem in (6) is equivalent to a generalized eigenvalue problem. Let $C_{xy} = X^T S Y$, $C_{xx} = X^T X$, and $C_{yy} = Y^T Y$. Then the optimal solution of W_x is the eigenvectors corresponding to the c largest eigenvalues of $C_{xy} C_{yy}^{-1} C_{xy}^T W_x = \Lambda^2 C_{xx} W_x$, and the optimal solution of W_y can be obtained by $W_y = C_{yy}^{-1} C_{xy}^T W_x \Lambda^{-1}$.

Sequential Learning for Non-Orthogonal Projection

The above solution obtained by direct eigen-decomposition leads to a practical problem. In real world datasets, most of the variance is contained in a few top projections. The orthogonality constraints force the solution to pick directions with low variance progressively. Since the variances of different projected dimensions are different and larger-variance projected dimensions carry more information, using each eigenvector to generate one bit in hash code is not reasonable (Kong and Li 2012b). To tackle this issue, (Gong and Lazebnik 2011) and (Kong and Li 2012b) proposed orthogonal rotation learning methods to reduce the quantization error while preserving the orthogonality constraints. But those methods focus on unsupervised unimodal learning which doesn't fit for our settings. It has been verified that the projection vectors that are not necessarily orthogonal to each other might achieve better performance than orthogonal projection vectors in practice (Wang, Kumar, and Chang 2012).

Motivated by this, we propose a sequential strategy to learn the hashing function bit by bit without imposing the orthogonality constraints.

Note that we aim to reconstruct the similarity matrix by the learned hash codes. Assuming that the projection vectors $w_x^{(1)}, ..., w_x^{(t-1)}$ and $w_y^{(1)}, ..., w_y^{(t-1)}$ have been learned, we then need to learn the next projection vectors $w_x^{(t)}$ and $w_y^{(t)}$. Let us define a residue matrix R_t as follows:

$$R_t = cS - \sum_{k=1}^{t-1} sgn(Xw_x^{(k)}) sgn(Yw_y^{(k)})^T. \quad (7)$$

Based on our original objective function in (4), to learn the best projection vectors $w_x^{(t)}$ and $w_y^{(t)}$ after the previous projection vectors $w_x^{(1)}, ..., w_x^{(t-1)}$ and $w_y^{(1)}, ..., w_y^{(t-1)}$ have been learned, our objective function can be written as follows:

$$\min_{w_x^{(t)}, w_y^{(t)}} \left\| sgn(Xw_x^{(t)}) sgn(Yw_y^{(t)})^T - R_t \right\|_F^2. \quad (8)$$

Similar to the problem in (6), we can apply the spectral relaxation trick to the objective function in (8) and obtain the one bit projection vectors by solving a generalized eigenvalue problem which can be got by substituting $w_x^{(t)}$, $w_y^{(t)}$, and R_t for W_x, W_y, and S in (6). Here, the semantic correlation is defined as follows:

$$\begin{aligned} C_{xy}^{(t)} &= X^T R_t Y \\ &= cX^T SY - \sum_{k=1}^{t-1} X^T sgn(Xw_x^{(k)}) sgn(Yw_y^{(k)})^T Y \\ &= C_{xy}^{(t-1)} - X^T sgn(Xw_x^{(t-1)}) sgn(Yw_y^{(t-1)})^T Y, \end{aligned}$$

which can also be efficiently calculated in an incremental mode with time complexity linear to n.

The overall sequential learning algorithm of our SCM hashing method is briefly summarized in Algorithm 1, where $C_{xy}^{(0)} = X^T SY = 2(X^T \tilde{L})(Y^T \tilde{L})^T - (X^T \mathbf{1}_n)(Y^T \mathbf{1}_n)^T$, and γ is a very small positive number 10^{-6} to overcome numerical problems.

Please note that although all the supervised information in the $n \times n$ matrices $\{R_t | t = 0, \ldots, c\}$ can be fully used in our algorithm, we elegantly avoid *explicitly* computing the pairwise similarity matrices. This has dramatically reduced the training time complexity from $O(n^2)$ to $O(n)$. Furthermore, it is easy to find the nice property of SCM that the solution of the hash function for each bit has a closed-form solution and no hyper-parameters or stopping conditions are needed for tuning during learning.

Algorithm 1 Sequential Learning Algorithm for SCM.

Input:
X, Y - feature vectors of the multimodal data
$\tilde{L}$ - normalized semantic lables
c - code length
Output:
$W_x = [w_x^{(1)}, w_x^{(2)}, ..., w_x^{(c)}]$
$W_y = [w_y^{(1)}, w_y^{(2)}, ..., w_y^{(c)}]$
Procedure:
$C_{xy}^{(0)} \leftarrow 2(X^T \tilde{L})(Y^T \tilde{L})^T - (X^T \mathbf{1}_n)(Y^T \mathbf{1}_n)^T$;
$C_{xy}^{(1)} \leftarrow c \times C_{xy}^{(0)}$;
$C_{xx} \leftarrow X^T X + \gamma I_{d_x}$;
$C_{yy} \leftarrow Y^T Y + \gamma I_{d_y}$;
for $t = 1 \to c$ **do**
  Solving the following generalized eigenvalue problem $C_{xy}^{(t)} C_{yy}^{-1} [C_{xy}^{(t)}]^T w_x = \lambda^2 C_{xx} w_x$,
  we can obtain the optimal solution $w_x^{(t)}$ corresponding to the largest eigenvalue λ_{max};
  $w_y^{(t)} \leftarrow \frac{C_{yy}^{-1} C_{xy}^T w_x^{(t)}}{\lambda_{max}}$;
  $h_x^{(t)} \leftarrow sgn(Xw_x^{(t)})$;
  $h_y^{(t)} \leftarrow sgn(Yw_y^{(t)})$;
  $U = (X^T sgn(Xw_x^{(t)}))(Y^T sgn(Yw_y^{(t)}))^T$;
  $C_{xy}^{(t+1)} \leftarrow C_{xy}^{(t)} - U$;
end for

Complexity Analysis

During the training procedure for sequential learning, the computational cost for initializing the C_{xy}, C_{xx} and C_{yy} is $O(d_x mn + d_y mn + d_x d_y m + {d_x}^2 n + {d_y}^2 n)$ in total. To learn each bit, solving the generalized eigenvalue problem takes $O({d_x}^3 + {d_y}^3 + d_x {d_y}^2 + {d_x}^2 d_y)$ time, which is independent of the size of training set n. The time for updating C_{xy} is $O(d_x n + d_y n + d_x d_y)$. Hence, the total time complexity for sequential learning is $O(d_x d_y m + c({d_x}^3 + {d_y}^3 + d_x {d_y}^2 + {d_x}^2 d_y) + (d_x m + d_y m + {d_x}^2 + {d_y}^2 + c d_x + c d_y) n)$.

For orthogonal projection learning, the computation is simpler. All c projections can be learned in one pass by solving only one generalized eigenvalue problem. Then the time complexity is $O(d_x d_y m + {d_x}^3 + {d_y}^3 + d_x {d_y}^2 + {d_x}^2 d_y + (d_x m + d_y m + {d_x}^2 + {d_y}^2) n)$. Typically, d_x, d_y and m will be much less than n.

Hence, the training time complexity of both the orthogonal projection and the sequential (non-orthogonal projection) learning methods is linear to the size of the training set. Moreover, the computational bottleneck in our algorithm is from matrix multiplication, which can be easily parallelized.

During the query procedure, the computational cost of encoding a query point with our learned hashing function is $O(cd_x)$ or $O(cd_y)$. Hence, the query time complexity of our SCM method is also very low.

The training time complexity of other existing methods, such as CVH (Kumar and Udupa 2011), CRH (Zhen and Yeung 2012a) and MLBE (Zhen and Yeung 2012b), is at least $O(n^2)$ because they have to compute all the elements in the $n \times n$ similarity matrix if all the supervised information need to be used. Hence, our SCM is much more scalable than existing SMH methods, which will be verified by the following experimental results.

Experiment

In this section, experimental results on two real-world multimodal multimedia datasets are used to verify the effectiveness of our SCM hashing method. All our experiments are conducted on a workstation with Intel(R) Xeon(R) CPU X7560@2.27GHz and 64 GB RAM.

Datasets

Two widely used datasets are adopted for evaluation. One is the NUS-WIDE dataset (Chua et al. 2009), and the other is the Wiki dataset (Rasiwasia et al. 2010).

NUS-WIDE (Chua et al. 2009) is a public image dataset containing 269,648 images crawled from Flickr, together with the associated raw tags of these images. Furthermore, the semantic labels of 91 concepts (categories) for these images are also available in the dataset. We select 186,577 image-tag pairs that belong to the 10 largest concepts. The images are represented by 500-dimensional bag-of-visual words (BOVW) and the tags are represented by 1000-dimensional tag occurrence feature vectors.

The Wiki dataset (Rasiwasia et al. 2010) is crawled from the Wikipedia's *featured articles*. It consists of 2,866 documents which are image-text pairs and annotated with semantic labels of 10 categories. Each image is represented by a 128-dimensional bag-of-visual SIFT feature vector and each text is represented by a 10-dimensional feature vector generated by latent Dirichlet allocation (Blei, Ng, and Jordan 2001).

Baselines and Evaluation Scheme

The most typical task for SMH is cross-modal retrieval. In our experiment, we evaluate our method on two cross-modal retrieval tasks: querying image database by text keywords, and querying text database by image examples. We compare our SCM method with several state-of-the-art multimodal hashing methods, including CCA (Gong and Lazebnik 2011), CVH (Kumar and Udupa 2011), CRH (Zhen and Yeung 2012a), and MLBE (Zhen and Yeung 2012b). [1] Among them, CCA is an unsupervised method and all the other methods are supervised. To integrate the semantic labels into CCA, we also evaluate a 3-view CCA method which regards label vectors as the third modality. We denote this method as CCA-3V. The orthogonal projection learning method and the sequential learning method of SCM are abbreviated as SCM-Orth and SCM-Seq, respectively.

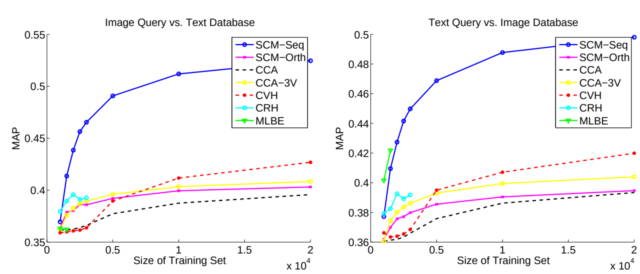
As in most existing SMH methods, the accuracy is evaluated by Mean Average Precision (MAP) (Zhen and Yeung 2012a). For a query q, the average precision (AP) is defined as $AP(q) = \frac{1}{L_q}\sum_{r=1}^{n} P_q(r)\delta_q(r)$, where L_q is the number of ground-truth neighbors of query q in database (training set), 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 rth retrieved entity is a ground-truth neighbor and $\delta_q(r) = 0$ otherwise. Ground-truth neighbors are defined as those pairs of entities (image or text) which share at least one semantic label. Given a query set of size Q, the MAP is defined as the mean of the average precision scores for all the queries in the query set: $MAP = \frac{1}{Q}\sum_{i=1}^{Q} AP(q_i)$.

In some settings of the following experiments, a random set of entities should be sampled from the whole database as training set. In such cases, five rounds of experiments are performed and the average MAP score is reported.

Scalability

To investigate the scalability of different methods, we evaluate the training time of different methods on NUS-WIDE dataset by varying the size of training set from 500 to 20,000. The code length is fixed to 16 in this experiment. The training time is reported in Table 1, where '-' denotes an untested value which is obvious to be relatively large and unnecessary to be tested. From Table 1, it is easy to find that the training time complexity of CVH, CRH and MLBE is much higher than that of our SCM-Seq and SCM-Orth methods when the size of the training set is relatively large, e.g., larger than 10,000. Because the motivation of hashing is to solve large-scale similarity search problems, the size of training set in most real hashing applications will be larger than tens of thousand. Hence, it is obvious that CVH, CRH and MLBE are not scalable, but both our SCM-Seq and SCM-Orth can be easily scaled up to large-scale applications. Note that CCA is an unsupervised method, and CCA-3V is a naive supervised method which cannot effectively utilize the supervised information. We list the results of CCA and CCA-3V in the table just for reference.

Figure 1 reports the MAP results on NUS-WIDE dataset by varying the size of training set. We can find that in most cases, our SCM-Seq method achieves the best accuracy. The SCM-Orth method doesn't achieve desirable result due to the quantization loss. In some cases, some traditional supervised methods, such as MLBE, can achieve promising results (refer to Figure 1(b)) when the available training set is small. However, as the available training set increases, most traditional supervised methods cannot fully utilize the available information for training due to high time complexity. On the contrary, our SCM can fully utilize the available information to further improve the accuracy as more and more training points are available.



(a) (b)

Figure 1: MAP on NUS-WIDE dataset by varying the size of training set.

[1]Since the original implementation of CVH is not publicly available, we implement it by ourselves. The source codes of all the other methods are kindly provided by the authors.

Table 1: Training time (in seconds) on NUS-WIDE dataset by varying the size of training set.

Method\Size of Training Set	500	1000	1500	2000	2500	3000	5000	10000	20000
SCM-Seq	276	249	303	222	236	260	248	228	230
SCM-Orth	36	80	85	77	83	76	110	87	102
CCA	25	20	23	22	25	22	28	38	44
CCA-3V	69	57	68	69	62	55	67	70	86
CVH	62	116	123	149	155	170	237	774	1630
CRH	68	253	312	515	760	1076	-	-	-
MLBE	67071	126431	-	-	-	-	-	-	-

Accuracy

Accuracy on NUS-WIDE Dataset The whole NUS-WIDE dataset contains 186,577 points. We use 99% of the data as the training set (database) and the remaining 1% to form the query set. Since the whole training set is too large, some baseline methods cannot be trained using the whole database. We conduct two experiments for evaluation: one is with a small-scale training set of 2,000 entities sampled from the database, and the other is with large-scale training set containing the whole database. Table 2 and Table 3 report the MAP results of these two experiments, respectively. We can find that our SCM-Seq method can outperform other baselines in all the cases.

Table 2: MAP results on small-scale training set of NUS-WIDE. The best performance is shown in boldface.

Task	Method	Code Length		
		$c = 16$	$c = 24$	$c = 32$
Image Query v.s. Text Database	SCM-Seq	**0.4385**	**0.4397**	**0.4390**
	SCM-Orth	0.3804	0.3746	0.3662
	CCA	0.3625	0.3586	0.3565
	CCA-3V	0.3826	0.3741	0.3692
	CVH	0.3608	0.3575	0.3562
	CRH	0.3957	0.3965	0.3970
	MLBE	0.3697	0.3620	0.3540
Text Query v.s. Image Database	SCM-Seq	**0.4273**	**0.4265**	**0.4259**
	SCM-Orth	0.3757	0.3625	0.3581
	CCA	0.3619	0.3580	0.3560
	CCA-3V	0.3801	0.3721	0.3676
	CVH	0.3640	0.3596	0.3581
	CRH	0.3926	0.3910	0.3904
	MLBE	0.3877	0.3636	0.3551

Table 3: MAP results on large-scale training set of NUS-WIDE. The best performance is shown in boldface.

Task	Method	Code Length		
		$c = 16$	$c = 24$	$c = 32$
Image Query v.s. Text Database	SCM-Seq	**0.5451**	**0.5501**	**0.5412**
	SCM-Orth	0.4146	0.3886	0.3890
	CCA	0.4078	0.3964	0.3886
	CCA-3V	0.4132	0.3980	0.3895
Text Query v.s. Image Database	SCM-Seq	**0.5147**	**0.5153**	**0.5105**
	SCM-Orth	0.4043	0.3788	0.3676
	CCA	0.4038	0.3934	0.3861
	CCA-3V	0.4088	0.3954	0.3877

Accuracy on Wiki Dataset For the Wiki dataset, we use 80% of the data as the training set and the remaining 20% to form the query set.

The MAP results are reported in Table 4 with various code lengths. Once again, the experimental results show that our SCM-Seq method can achieve much better accuracy than baseline methods.

Table 4: MAP results on Wiki dataset. The best performance is shown in boldface.

Task	Method	Code Length		
		$c = 16$	$c = 24$	$c = 32$
Image Query v.s. Text Database	SCM-Seq	**0.2393**	**0.2379**	**0.2419**
	SCM-Orth	0.1549	0.1545	0.1550
	CCA	0.1805	0.1656	0.1618
	CCA-3V	0.1713	0.1651	0.1653
	CVH	0.1499	0.1408	0.1372
	CRH	0.1586	0.1618	0.1578
	MLBE	0.2015	0.2238	0.2342
Text Query v.s. Image Database	SCM-Seq	**0.2325**	**0.2454**	**0.2452**
	SCM-Orth	0.1470	0.1370	0.1284
	CCA	0.1566	0.1398	0.1317
	CCA-3V	0.1544	0.1426	0.1397
	CVH	0.1315	0.1171	0.1080
	CRH	0.1293	0.1276	0.1225
	MLBE	0.2000	0.2384	0.2186

Conclusion

Most existing supervised multimodal hashing methods are not scalable. In this paper, by avoiding explicitly computing the semantic similarity matrix, we have proposed a very effective supervised multimodal hashing method, called SCM, with high scalability. Furthermore, our SCM method has a nice property that no hyper-parameters or stopping conditions are needed for tuning during learning. Experiments on real datasets have demonstrated that our method with sequential learning can significantly outperform the state-of-the-art methods in terms of both accuracy and scalability.

Acknowledgements

This work is supported by the NSFC (No. 61100125), the 863 Program of China (No. 2012AA011003), and the Program for Changjiang Scholars and Innovative Research Team in University of China (IRT1158, PCSIRT).

References

Andoni, A., and Indyk, P. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM* 51(1):117–122.

Barnard, K., and Forsyth, D. A. 2001. Learning the semantics of words and pictures. In *ICCV*, 408–415.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2001. Latent dirichlet allocation. In *NIPS*, 601–608.

Bronstein, M. M.; Bronstein, A. M.; Michel, F.; and Paragios, N. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, 3594–3601.

Chen, N.; Zhu, J.; Sun, F.; and Xing, E. P. 2012. Large-margin predictive latent subspace learning for multiview data analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(12):2365–2378.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*.

Dean, T. L.; Ruzon, M. A.; Segal, M.; Shlens, J.; Vijayanarasimhan, S.; and Yagnik, J. 2013. Fast, accurate detection of 100, 000 object classes on a single machine. In *CVPR*, 1814–1821.

Ge, T.; He, K.; Ke, Q.; and Sun, J. 2013. Optimized product quantization for approximate nearest neighbor search. In *CVPR*, 2946–2953.

Gong, Y., and Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, 817–824.

Gong, Y.; Ke, Q.; Isard, M.; and Lazebnik, S. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision* 106(2):210–233.

He, J.; Feng, J.; Liu, X.; Cheng, T.; Lin, T.-H.; Chung, H.; and Chang, S.-F. 2012. Mobile product search with bag of hash bits and boundary reranking. In *CVPR*, 3005–3012.

Heo, J.-P.; Lee, Y.; He, J.; Chang, S.-F.; and Yoon, S.-E. 2012. Spherical hashing. In *CVPR*, 2957–2964.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3/4):321–377.

Hwang, S. J., and Grauman, K. 2012. Learning the relative importance of objects from tagged images for retrieval and cross-modal search. *International Journal of Computer Vision* 100(2):134–153.

Kong, W., and Li, W.-J. 2012a. Double-bit quantization for hashing. In *AAAI*.

Kong, W., and Li, W.-J. 2012b. Isotropic hashing. In *NIPS*, 1655–1663.

Kong, W.; Li, W.-J.; and Guo, M. 2012. Manhattan hashing for large-scale image retrieval. In *SIGIR*, 45–54.

Kulis, B., and Darrell, T. 2009. Learning to hash with binary reconstructive embeddings. In *NIPS*, 1042–1050.

Kumar, S., and Udupa, R. 2011. Learning hash functions for cross-view similarity search. In *IJCAI*, 1360–1365.

Lin, R.-S.; Ross, D. A.; and Yagnik, J. 2010. Spec hashing: Similarity preserving algorithm for entropy-based coding. In *CVPR*, 848–854.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012a. Supervised hashing with kernels. In *CVPR*, 2074–2081.

Liu, X.; He, J.; Liu, D.; and Lang, B. 2012b. Compact kernel hashing with multiple features. In *ACM Multimedia*, 881–884.

Neyshabur, B.; Srebro, N.; Salakhutdinov, R.; Makarychev, Y.; and Yadollahpour, P. 2013. The power of asymmetry in binary hashing. In *NIPS*, 2823–2831.

Norouzi, M., and Fleet, D. J. 2011. Minimal loss hashing for compact binary codes. In *ICML*, 353–360.

Norouzi, M.; Fleet, D. J.; and Salakhutdinov, R. 2012. Hamming distance metric learning. In *NIPS*, 1070–1078.

Ou, M.; Cui, P.; Wang, F.; Wang, J.; Zhu, W.; and Yang, S. 2013. Comparing apples to oranges: a scalable solution with heterogeneous hashing. In *KDD*, 230–238.

Rasiwasia, N.; Pereira, J. C.; Coviello, E.; Doyle, G.; Lanckriet, G. R. G.; Levy, R.; and Vasconcelos, N. 2010. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*, 251–260.

Rastegari, M.; Choi, J.; Fakhraei, S.; III, H. D.; and Davis, L. S. 2013. Predictable dual-view hashing. In *ICML*, 1328–1336.

Salakhutdinov, R., and Hinton, G. E. 2009. Semantic hashing. *Int. J. Approx. Reasoning* 50(7):969–978.

Sharma, A.; Kumar, A.; III, H. D.; and Jacobs, D. W. 2012. Generalized multiview analysis: A discriminative latent space. In *CVPR*, 2160–2167.

Song, J.; Yang, Y.; Huang, Z.; Shen, H. T.; and Hong, R. 2011. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM Multimedia*, 423–432.

Song, J.; Yang, Y.; Yang, Y.; Huang, Z.; and Shen, H. T. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD Conference*, 785–796.

Strecha, C.; Bronstein, A. A.; Bronstein, M. M.; and Fua, P. 2012. LDAHash: Improved matching with smaller descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(1):66–78.

Torralba, A.; Fergus, R.; and Weiss, Y. 2008. Small codes and large image databases for recognition. In *CVPR*.

Tseng, K.-Y.; Lin, Y.-L.; Chen, Y.-H.; and Hsu, W. H. 2012. Sketch-based image retrieval on mobile devices using compact hash bits. In *ACM Multimedia*, 913–916.

Wang, J.; Wang, J.; Yu, N.; and Li, S. 2013. Order preserving hashing for approximate nearest neighbor search. In *ACM Multimedia*, 133–142.

Wang, J.; Kumar, S.; and Chang, S.-F. 2010. Sequential projection learning for hashing with compact codes. In *ICML*, 1127–1134.

Wang, J.; Kumar, S.; and Chang, S.-F. 2012. Semi-supervised hashing for large-scale search. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(12):2393–2406.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *NIPS*, 1753–1760.

Wu, F.; Yu, Z.; Yang, Y.; Tang, S.; Zhang, Y.; and Zhuang, Y. 2014. Sparse multi-modal hashing. *IEEE Transactions on Multimedia* 16(2):427–439.

Xu, B.; Bu, J.; Lin, Y.; Chen, C.; He, X.; and Cai, D. 2013. Harmonious hashing. In *IJCAI*.

Zhai, D.; Chang, H.; Zhen, Y.; Liu, X.; Chen, X.; and Gao, W. 2013. Parametric local multimodal hashing for cross-view similarity search. In *IJCAI*.

Zhang, P.; Zhang, W.; Li, W.-J.; and Guo, M. 2014. Supervised hashing with latent factor models. In *SIGIR*.

Zhang, D.; Wang, F.; and Si, L. 2011. Composite hashing with multiple information sources. In *SIGIR*, 225–234.

Zhen, Y., and Yeung, D.-Y. 2012a. Co-regularized hashing for multimodal data. In *NIPS*, 1385–1393.

Zhen, Y., and Yeung, D.-Y. 2012b. A probabilistic model for multimodal hash function learning. In *KDD*, 940–948.