Latent Semantic Sparse Hashing for Cross-Modal Similarity Search

Jile Zhou
Intelligence Multimedia Group
School of Software
Tsinghua University
Beijing, China
zhoujile539@gmail.com

Guiguang Ding
Intelligence Multimedia Group
School of Software
Tsinghua University
Beijing, China
dinggg@tsinghua.edu.cn

Yuchen Guo
Intelligence Multimedia Group
School of Software
Tsinghua University
Beijing, China
yuchen.w.guo@gmail.com

ABSTRACT

Similarity search methods based on hashing for effective and efficient cross-modal retrieval on large-scale multimedia databases with massive text and images have attracted considerable attention. The core problem of cross-modal hashing is how to effectively construct *correlation* between multi-modal representations which are heterogeneous intrinsically in the process of hash function learning. Analogous to Canonical Correlation Analysis (CCA), most existing cross-modal hash methods embed the heterogeneous data into a joint abstraction space by linear projections. However, these methods fail to bridge the semantic gap more effectively, and capture high-level latent semantic information which has been proved that it can lead to better performance for image retrieval. To address these challenges, in this paper, we propose a novel Latent Semantic Sparse Hashing (LSSH) to perform cross-modal similarity search by employing Sparse Coding and Matrix Factorization. In particular, LSSH uses Sparse Coding to capture the salient structures of images, and Matrix Factorization to learn the latent concepts from text. Then the learned latent semantic features are mapped to a joint abstraction space. Moreover, an iterative strategy is applied to derive optimal solutions efficiently, and it helps LSSH to explore the correlation between multi-modal representations efficiently and automatically. Finally, the unified hashcodes are generated through the high level abstraction space by quantization. Extensive experiments on three different datasets highlight the advantage of our method under cross-modal scenarios and show that LSSH significantly outperforms several state-of-the-art methods.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

Hashing, Cross-Modal Retrieval, Heterogeneous Data Sources, Correlation, Sparse Coding, Matrix Factorization

1. INTRODUCTION

Similarity or Nearest Neighbor (NN) search, a method of searching semantically related results from a collection of objects for a query, lays the foundation for many important applications, such as information retrieval, data mining, and computer vision. Hashing-based methods [10, 6], one of the most well-known *Approximate Nearest Neighbor search* (ANN) methods, has garnered considerable interest in recent years for their great efficiency gains in massive data. The goal of hashing is to learn binary-code representation for data while preserving the similarity structure in the original feature space. One of the most famous models, locality-sensitive hashing (LSH) [1], which employs random linear projections to map feature vectors to binary codes, is quite efficient in both space and time. However, LSH may lead to ineffective codes in practice because it is data-independent [34]. Several machine learning techniques are used to design more effective hashing to overcome this problem, such as Boosting algorithm, Restricted Boltzmann Machines, Manifold Learning, Supervised Learning, Kernel Learning and PCA, which respectively generate Parameter sensitive Hashing [26], Semantic Hashing [25], Spectral Hashing [30], Supervised Hashing [16], Kernelized Hashing [13] and PCA Hashing [29]. Moreover, several literatures take the quantization of Hamming space into account, and have achieved superior results, such as K-means Hashing [8], ITQ Hashing [7] and Double-Bit Hashing [12].

Most existing hashing methods can only be applied to unimodal data. However, with the fast growth of multimedia content on the Web, like Wikipedia, Flickr and Twitter, the cross-modal retrieval problem, returning similar results of all modals for a given query, have attracted increasing attention and more studies about it emerge. Taking Wikipedia as example, it contains images and text. When a query word or picture is given, the system should return both relevant articles and images. This is central to many applications of practical interest [23]. However, designing effective and efficient hashing methods over heterogeneous cross-modal datasets is still remaining as an open issue.

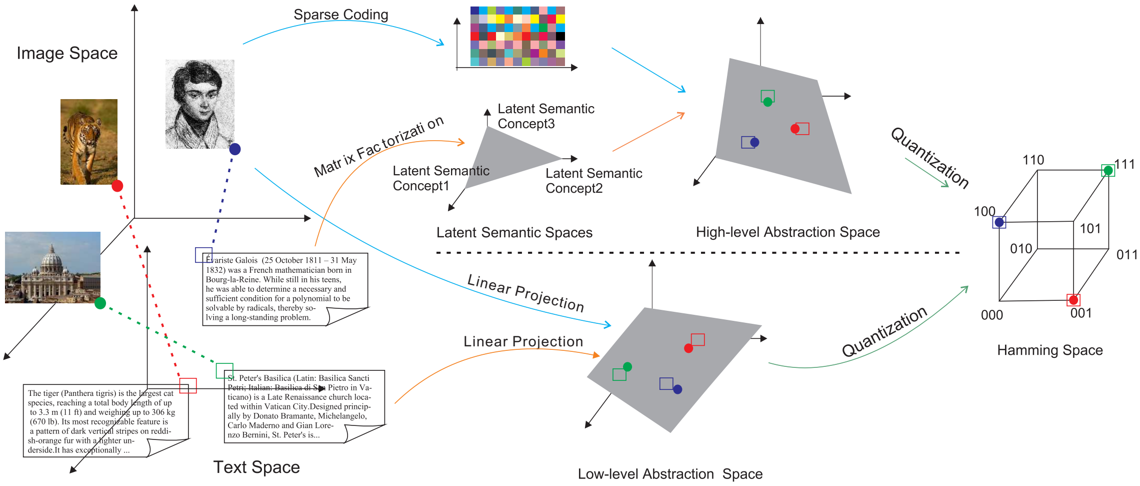
Image Space
Sparse Coding
Matr ix Fac torizati on
Latent Semantic Concept3
Latent Semantic Concept1
Latent Semantic Concept2
Latent Semantic Spaces
High-level Abstraction Space
Quantization
110
111
100
101
010
011
000
001
Hamming Space
Linear Projection
Linear Projection
Quantization
Évariste Galois (25 October 1811 – 31 May 1832) was a French mathematician born in Bourg-la-Reine. While still in his teens, he was able to determine a necessary and sufficient condition for a polynomial to be solvable by radicals, thereby solving a long-standing problem.
The tiger (Panthera tigris) is the largest cat species, reaching a total body length of up to 3.3 m (11 ft) and weighing up to 306 kg (670 lb). Its most recognizable feature is a pattern of dark vertical stripes on reddish-orange fur with a lighter underside.It has exceptionally ...
St. Peter's Basilica (Latin: Basilica Sancti Petri; Italian: Basilica di San Pietro in Vaticano) is a Late Renaissance church located within Vatican City.Designed principally by Donato Bramante, Michelangelo, Carlo Maderno and Gian Lorenzo Bernini, St. Peter's is...
Text Space
Low-level Abstraction Space

Figure 1: The difference between the proposed LSSH and existing cross-modal hashing, illustrated with toy data. Up) LSSH maps the text and images from their respective natural spaces to two isomorphic latent semantic spaces firstly, then projects the semantic spaces to a joint high level abstraction space. The latent semantic spaces are learned using sparse coding and matrix factorization respectively. Bottom) Existing cross-view models map the text and images to a joint low level abstraction space directly. At last, the learned abstraction space is quantized to the Hamming space in all hashing methods.

The core problem of cross-modal hash function learning (HFL) is how to construct correlation between multi-modal representations which are heterogeneous intrinsically in the process of HFL. Recently, a few studies designed new hashing techniques to index multi-modal data into a common Hamming space [33, 14, 11, 3, 36, 27]. As shown in Figure 2, analogous to Canonical Correlation Analysis (CCA) [9], these models find the linear projections to embed the heterogeneous data into a joint abstraction space while maximizing the cross-correlation between images and text on a training set. Then a quantization rule is applied to map the abstraction representations to binary hash codes. In complex situations, i.e. the semantic gap between multi-modal data (e.g. visual features and text features) is large, however, these models do not extract useful joint features because they fail to capture the common latent information. Therefor, they fail to generate effective hashcodes when dealing with complex multi-modal data.

Prior works have shown that the model which combines semantic abstraction for both images and text with explicit modeling of cross-correlations in a joint space can achieve better results for cross-multimedia retrieval [19, 23, 24]. Motivated by this observation, we propose a novel Latent Semantic Sparse Hashing (LSSH) algorithm to learn binary codes for multimedia data sources with text and images. As illustrated in Figure 1. up, LSSH represents text and image features in a new latent semantic space respectively, in which the heterogeneous representations of the same topic will show more common properties [23, 24]. In fact, LSSH uses Sparse Coding (SC) to capture the salient structures (e.g. edges) of images, and Matrix Factorization (MF) to learn the latent concepts from text. Then the learned latent semantic features are mapped to a joint abstraction space. Furthermore, an iterative strategy is applied to derive optimal solutions, and it helps LSSH to explore the cross-correlation between multi-modal representations efficiently and automatically in the process of HFL. Finally, the unified hashcodes are generated from high level abstraction space by quantization. The contributions of LSSH can be summarized as follows:

1. We propose a novel cross-modal hashing framework to efficiently construct the correlations between heterogeneous data. Moreover, the proposed method utilizes SC and MF to merge multiple latent semantic descriptions to generate discriminate binary codes.

2. An iterative strategy is used to help LSSH explore the cross-correlation between multi-modal representations efficiently and automatically.

3. Extensive experiments on three datasets highlight the advantages of LSSH under cross-view scenarios and show that LSSH significantly outperforms several state-of-the-art methods. Especially, LSSH shows significant improvement for cross-modal retrieval with long codes.

The rest of this paper is organized as follows. We formulate several related cross-modal hashing methods and Canonical Correlation Analysis (CCA) within the same framework in Section 2. Section 3 presents our proposed method. Section 4 provides extensive experimental validation on three datasets. The conclusions are given in Section 5.

2. RELATED WORK

In this section, we show that a variety of cross-modal methods (CMH), including CCA [9], Data Fusion Hashing (DFH) [3], and Cross-View Hashing (CVH) [14] can be formulated within the framework of *correlation analysis* where correlation is used as the objective function. Obviously, correlation of heterogeneous features is directly related to the empirical ANN performance for cross-modal retrieval tasks.

2.1 Canonical Correlation Analysis

Consider random vector of the form ($\mathbf{x}$,$\mathbf{y}$) (i.e. image and text feature), and the given samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$. CCA project $\mathbf{x}$ respectively $\mathbf{y}$ onto directions $\mathbf{w}_x$ and $\mathbf{w}_y$:

$$\mathbf{x} \rightarrow \mathbf{w}_x^T \mathbf{x}, \quad \mathbf{y} \rightarrow \mathbf{w}_y^T \mathbf{y}$$

then maximise the correlation between the two modalities which can be defined as follows:

$$\begin{aligned}&\max_{\mathbf{w}_x,\mathbf{w}_y} \mathbf{w}_x^T \widehat{\mathbb{E}}_\Lambda[\mathbf{x}\mathbf{y}^T]\mathbf{w}_y \\ &s.t. \mathbf{w}_x^T\widehat{\mathbb{E}}_\Lambda[\mathbf{x}\mathbf{x}^T]\mathbf{w}_x = 1, \mathbf{w}_y^T\widehat{\mathbb{E}}_\Lambda[\mathbf{y}\mathbf{y}^T]\mathbf{w}_y = 1\end{aligned} \tag{1}$$

where Λ is a diagonal matrix whose diagonal entry $\Lambda_{ii} = 1/n$, and $\widehat{\mathbb{E}}_P[f(\mathbf{x},\mathbf{y})]$ denotes the weighted empirical expectation of the function $f(\mathbf{x},\mathbf{y})$, which is computed by the following equation

$$\widehat{\mathbb{E}}_P[f(\mathbf{x},\mathbf{y})] = \sum_{i,j=1}^{n} \mathbf{P}_{ij} f(\mathbf{x}_i,\mathbf{y}_j) \tag{2}$$

The optimization of (1) can be solved as a generalized eigenvalue problem (GEV) :

$$\begin{bmatrix}\mathbf{0} & \widehat{\mathbb{E}}_\Lambda[\mathbf{x}\mathbf{y}^T] \\ \widehat{\mathbb{E}}_\Lambda[\mathbf{y}\mathbf{x}^T] & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{w}_x \\ \mathbf{w}_y\end{bmatrix} = \lambda\begin{bmatrix}\widehat{\mathbb{E}}_\Lambda[\mathbf{x}\mathbf{x}^T] & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbb{E}}_\Lambda[\mathbf{y}\mathbf{y}^T]\end{bmatrix}\begin{bmatrix}\mathbf{w}_x \\ \mathbf{w}_y\end{bmatrix}$$

where λ is the generalized eigenvalue.

2.2 Data Fusion Hashing

DFH [3] embeds the input data from two arbitrary spaces into the Hamming space in a supervised way. Given sample pair $(\mathbf{x}_i, \mathbf{y}_i)$ and similarity label $s_i \in \{+1, -1\}$, DFH maximizing:

$$\sum_i s_i sign(\mathbf{w}_x^T\mathbf{x}_i) sign(\mathbf{w}_y^T\mathbf{y}_i) \tag{3}$$

where $sign(u) = -1$ if $u < 0$, or 1 otherwise, $\forall u \in \mathbb{R}$ is sign function. Discarding the sign function, Equation (3) is closely related to a simpler correlation function:

$$\begin{aligned}&\max_{\mathbf{w}_x,\mathbf{w}_y} \mathbf{w}_x^T(\widehat{\mathbb{E}}_{\Lambda_+}[\mathbf{x}\mathbf{y}^T] - \widehat{\mathbb{E}}_{\Lambda_-}[\mathbf{x}\mathbf{y}^T])\mathbf{w}_y \\ &s.t. \mathbf{w}_x^T\mathbf{w}_x = 1, \mathbf{w}_y^T\mathbf{w}_y = 1\end{aligned} \tag{4}$$

the constraints are added to avoid trivial solutions, and Λ_+ is a diagonal matrix whose the i-th diagonal entry equals $1/|S_+|$ if $s_i = 1$, or 0 otherwise, where $S_+ = \{(\mathbf{x}_i, \mathbf{y}_i)|s_i = +1, \forall i\}$. The definition of Λ_- is analogous. And formula (4) can be solved by Singular Value Decomposition (SVD).

2.3 Cross-View Hashing

CVH [14] maximises the weighted cumulative correlation:

$$\begin{aligned}&\max_{\mathbf{w}_x,\mathbf{w}_y} 2\mathbf{w}_x^T\widehat{\mathbb{E}}_W[\mathbf{x}\mathbf{y}^T]\mathbf{w}_y^T - \mathbf{w}_x^T\widehat{\mathbb{E}}_{L'}[\mathbf{x}\mathbf{x}^T]\mathbf{w}_x^T - \mathbf{w}_y^T\widehat{\mathbb{E}}_{L'}[\mathbf{y}\mathbf{y}^T]\mathbf{w}_y^T \\ &s.t. \mathbf{w}_x^T\widehat{\mathbb{E}}_\Lambda[\mathbf{x}\mathbf{x}^T]\mathbf{w}_x = 1, \mathbf{w}_y^T\widehat{\mathbb{E}}_\Lambda[\mathbf{y}\mathbf{y}^T]\mathbf{w}_y = 1\end{aligned} \tag{5}$$

where $\mathbf{W}_{ij}$ be the similarity between instances i and j, $\mathbf{L}' = 2\mathbf{L} + \mathbf{D}$, $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D}_{ii} = \sum_i \mathbf{W}_{ij}$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix. Anyway, formula (5) can be transform to a GVE problem [14]:

$$\begin{bmatrix}-\widehat{\mathbb{E}}_{L'}[\mathbf{x}\mathbf{x}^T] & \widehat{\mathbb{E}}_W[\mathbf{x}\mathbf{y}^T] \\ \widehat{\mathbb{E}}_W[\mathbf{y}\mathbf{x}^T] & -\widehat{\mathbb{E}}_{L'}[\mathbf{y}\mathbf{y}^T]\end{bmatrix}\begin{bmatrix}\mathbf{w}_x \\ \mathbf{w}_y\end{bmatrix} = \lambda\begin{bmatrix}\widehat{\mathbb{E}}_I[\mathbf{x}\mathbf{x}^T] & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbb{E}}_I[\mathbf{y}\mathbf{y}^T]\end{bmatrix}\begin{bmatrix}\mathbf{w}_x \\ \mathbf{w}_y\end{bmatrix}$$

Actually, CCA can be viewed as a special case of CVH by setting $\mathbf{W} = \mathbf{I}$ [14].

All aforementioned cross-modal models assume that heterogeneous data can be embedded into a common abstraction space directly. However, the assumption may not fit into real world scenarios, especially when the semantic gap between multi-modal data (e.g. visual features and text features) is large, it may reduce the accuracy of cross-modal similarity search significantly. Moreover, prior works have shown that the high-level latent semantic information can lead to better performance for image retrieval and bridge the semantic gap more efficiently [19, 23, 24]. Hence, the proposed LSSH constructs correlation between two modalities in latent semantic spaces.

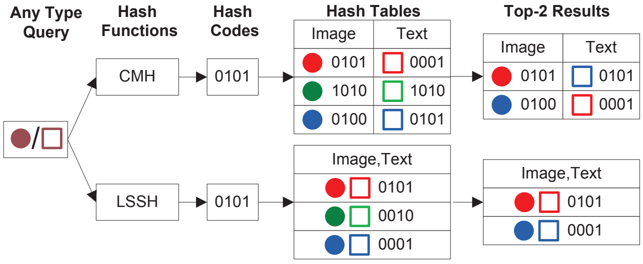

Figure 2: Flowchart of LSSH and existing CMH methods, illustrated with toy data. Up) Existing CMH methods learn independent hash codes for each modal of instances. Bottom) LSSH, an integrated hashing method for cross-modal, represents image and text feature by unified hashcodes.

3. LATENT SEMANTIC SPARSE HASHING FOR CROSS MODAL

In this section, we present a novel approach for cross-modal similarity search. We restrict the discussion to multimodal instances consisting of images and text as they are the most common and important scene in real world.

3.1 Model Formulation

Suppose that $\mathcal{O} = \{o_i\}_{i=1}^n$ is a set of multi-modal instances, which only consists of an image and its accompanying text, i.e. $o_i = (\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i \in \mathbb{R}^m$ is the m-dimensional image descriptor, and $\mathbf{y}_i \in \mathbb{R}^d$ is the d-dimensional text feature (usually, $m \neq d$). Given the codewords length k, the purpose of LSSH is to learn a integrated binary code which can bridge the semantic gap between heterogeneous data (i.e. image and text features) effectively while preserving the intrinsic similar structure of instances. As illustrated in Figure 2, queries of any type would be mapped to a common Hamming space according to related learned hash functions, which makes LSSH deal with queries with partial missing modalities. Scanning over the hash table linearly, the system returns similar results of all modalities for the given mapped query. CMH is quite efficient for online similarity search task, since only bit XOR operations are applied when calculating Hamming distance between binary codes. Moreover, compared with existing CMH, which learn independent hash codes for each modal of one instance, LSSH can cut down the online search time and the storage space of binary codes by half, while also promoting the retrieval precision significantly.

3.2 Latent Semantic Cross Correlation

Previous works have shown that the semantic modeling has at least two advantages for cross-modal retrieval [23, 24]. Firstly, it provides a high-level abstraction which can lead to substantially better performance for image retrieval. Secondly, the semantic spaces of heterogeneous data which describe the same instances are isomorphic. Motivated by these observations, LSSH constructs the cross-correlations in the latent semantic spaces.

As in Figure 1. up, we project the original image and text features to the latent semantic space respectively:

$$\mathcal{P}_I : \mathbb{R}^m \to \mathcal{S}_I^M, \quad \mathcal{P}_T : \mathbb{R}^d \to \mathcal{S}_T^D$$

where $\mathcal{P}_I$ and $\mathcal{P}_T$ denote the projections, and M and D is the dimension of $\mathcal{S}_I^M$ and $\mathcal{S}_T^D$ respectively. Then the isomorphic latent semantic features are mapped into a common high level abstraction space by linear projection, which is the simplest isomorphic function:

$$\mathbf{R}_I : \mathcal{S}_I^M \to \mathcal{A}^k, \quad \mathbf{R}_T : \mathcal{S}_T^D \to \mathcal{A}^k$$

where $\mathbf{R}_I \in \mathbb{R}^{M\times k}$ and $\mathbf{R}_T \in \mathbb{R}^{D\times k}$. In order to construct cross-correlation between two modalities, we require image and text features of the same instance to be equal in $\mathcal{A}^k$:

$$\mathbf{R}_I\mathcal{P}_I(\mathbf{x}_i) = \mathbf{R}_T\mathcal{P}_T(\mathbf{y}_i), \forall i \tag{6}$$

At last, binary hashcodes are obtained by the non-linear quantization function:

$$\mathcal{Q} : \mathcal{A}^k \to \mathcal{H}^k$$

where $\mathcal{H}^k$ is the k-dimension Hamming space. Several quantization algorithm has been proposed [17, 7, 12], however, this is not the focus of our research, hence, we simply regard the quantization function $\mathcal{Q}$ as a *sign* function.

3.3 Learning Latent Semantic Representation

Image Sparse Coding has been popularly used as an effective image representation in many applications, such as image classification [31], face recognition [32], image denoising [5] and image restoration [20]. The standard sparse coding, describing each sample using only several active vectors of dictionary, has at least two advantages for image representation. Firstly, the natural images may generally be described in terms of a small number of structural primitives [22], and the sparsity constraint in function (7) allows the learned representation to capture the salient structures. Secondly, the over-complete dictionary provides sufficient descriptive power for low-level features. Based on these observations, we use the Sparse Coding to capture the salient structures of images in LSSH. Let $\mathbf{X}$ be a set of m-dimensional image descriptors, i.e. $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in \mathbb{R}^{m\times n}$, the standard space coding with ℓ_1 regularization is:

$$\mathcal{O}_{\text{sc}}(\mathbf{B}, \mathbf{S}) = \|\mathbf{X} - \mathbf{BS}\|_F^2 + \sum_{i=1}^{n} \lambda|\mathbf{s}_i|_1 \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{m\times M}$ is the overcomplete basis set, i.e. $M > m$, $|| \cdot ||_F$ denotes Frobenius norm, and $\lambda > 0$ is the parameter to balance the reconstruction error and sparsity.

Text Matrix Factorization, as one of the most successful tools for learning the concepts or latent topics from text, has a wide range of applications in text mining and information retrieval. Let $\mathbf{Y}$ be a set of d-dimensional text descriptors, i.e. $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n] \in \mathbb{R}^{d\times n}$, LSSH learns the latent concepts by matrix factorization:

$$\mathcal{O}_{\text{mf}}(\mathbf{U}, \mathbf{A}) = \|\mathbf{Y} - \mathbf{UA}\|_F^2 \tag{8}$$

where $\mathbf{U} \in \mathbb{R}^{d\times D}$, $\mathbf{A} \in \mathbb{R}^{D\times n}$. In fact, each column vector $\mathbf{U}_{\cdot i}$ captures the higher-level features of original data, and each column vector $\mathbf{A}_{\cdot i}$ is the D-dimensional representation in latent semantic space [4].

3.4 Overall Objective Function

Let each bit of hashcodes represents a latent semantic concept in Matrix Factorization, i.e. $D = k$, then the formula (6) can be rewrite by left multiplication inverse of $\mathbf{R}_T$:

$$\mathbf{A}_{\cdot i} = \mathbf{R}_T^{-1}\mathbf{R}_I\mathbf{s}_i = \mathbf{R}\mathbf{s}_i, \forall i \tag{9}$$

where $\mathbf{R} = \mathbf{R}_T^{-1}\mathbf{R}_I$ is the linear projection. There is an intuitive interpretation about formula (9), that is a latent concept can be described by several salient structures from images. Moreover, we can approximate formula (9) by optimizing the cross-correlation:

$$\mathcal{O}_{\text{cc}}(\mathbf{R}) = \|\mathbf{A} - \mathbf{RS}\|_F^2 \tag{10}$$

The overall objective function, combining the Sparse Coding on image features $\mathcal{O}_{\text{sc}}$ given in formula (7), the Matrix Factorization on text features $\mathcal{O}_{\text{mf}}$ given in formula (8), and the cross-correlation between the latent semantic spaces $\mathcal{O}_{\text{cc}}$ given in formula (10), is written as below:

$$\begin{gathered}\min_{\mathbf{B},\mathbf{A},\mathbf{R},\mathbf{U},\mathbf{S}} \mathcal{O}(\mathbf{B}, \mathbf{A}, \mathbf{R}, \mathbf{U}, \mathbf{S}) = \mathcal{O}_{\text{sc}} + \mu\mathcal{O}_{\text{mf}} + \gamma\mathcal{O}_{\text{cc}} \\ s.t. \|\mathbf{B}_{\cdot i}\|^2 \leq 1, \|\mathbf{U}_{\cdot j}\|^2 \leq 1, \|\mathbf{R}_{\cdot t}\|^2 \leq 1, \forall i, j, t\end{gathered} \tag{11}$$

where $\mu > 0$ leverages the discrimination power of images and text latent representations, $\gamma > 0$ controls the linear connection of latent semantic spaces, and $\|\cdot\| \leq 1$ is typically applied to avoid trivial solutions.

3.5 Optimization Algorithm

The optimization problem (11) is non-convex with five matrices variables $\mathbf{B}, \mathbf{A}, \mathbf{R}, \mathbf{U}, \mathbf{S}$. Fortunately, it is convex with respect to any one of the five variables while fixing the other four. Therefore, the optimization problem can be solved by an iterative framework with the following listed steps until convergency.

Step1: Learn sparse representations $\mathbf{S}$ by fixing others variables, the problem (11) becomes

$$\begin{gathered}\min_{\mathbf{S}} ||\mathbf{X} - \mathbf{BS}||_F^2 + \sum_{i=1}^{n} \lambda|\mathbf{s}_i|_1 + \gamma||\mathbf{A} - \mathbf{RS}||_F^2 \\ \Leftrightarrow \min_{\mathbf{S}} || \begin{bmatrix} \mathbf{X} \\ \sqrt{\gamma}\mathbf{A} \end{bmatrix} - \begin{bmatrix} \mathbf{B} \\ \sqrt{\gamma}\mathbf{R} \end{bmatrix} \mathbf{S}||_F^2 + \sum_{i=1}^{n} \lambda|\mathbf{s}_i|_1\end{gathered} \tag{12}$$

We solve the problem (12) by using SLEP (Sparse Learning with Efficient Projections) package [1].

Step2: Learn latent semantic concepts $\mathbf{A}$ by fixing others variables, the problem (11) becomes

$$\min_{\mathbf{A}} \mu||\mathbf{Y} - \mathbf{UA}||_F^2 + \gamma||\mathbf{A} - \mathbf{RS}||_F^2 \tag{13}$$

By taking the derivative of formula (13) with respect to $\mathbf{A}$ and setting it to 0, we can obtain the close form solution:

$$\mathbf{A} = (\mathbf{U}^T\mathbf{U} + \frac{\gamma}{\mu}\mathbf{I})^{-1}(\frac{\gamma}{\mu}\mathbf{RS} + \mathbf{U}^T\mathbf{Y}) \tag{14}$$

where $\mathbf{I}$ is the identity matrix.

Step3: Learn $\mathbf{B}, \mathbf{R}, \mathbf{U}$ respectively using the Lagrange dual [15]. In fact, the learning problem of $\mathbf{B}, \mathbf{R}, \mathbf{U}$ is essentially same, hence we only show how to optimize $\mathbf{B}$. Fixing others variables, the problem (11) becomes the least squares problem with quadratic constraints:

$$\min_{\mathbf{B}} \|\mathbf{X} - \mathbf{BS}\|_F^2 \quad s.t. \|\mathbf{B}_{\cdot i}\|^2 \leq 1, \forall i \tag{15}$$

[1] http://parnec.nuaa.edu.cn/jliu/largeScaleSparseLearning.htm

Algorithm 1 Latent Semantic Sparse Hashing

```
Input:
    Image representation matrix $\mathbf{X}$ and text feature matrix $\mathbf{Y}$, parameters $\lambda, \mu, \gamma$ and hashcodes length $k$.
Output:
    Integrated hash codes $\mathbf{H}$, matrix variables $\mathbf{B}, \mathbf{U}, \mathbf{R}$.
Initialize $\mathbf{U}, \mathbf{A}, \mathbf{R}$ and $\mathbf{B}$ by random matrices respectively, and normalizing each column of $\mathbf{X}$ respectively $\mathbf{Y}$ by $\ell_2$ norm.
repeat
    Fix $\mathbf{U}, \mathbf{R}, \mathbf{B}$ and $\mathbf{A}$, update $\mathbf{S}$ as illustrated in Step1;
    Fix $\mathbf{U}, \mathbf{R}, \mathbf{B}$ and $\mathbf{S}$, update $\mathbf{A}$ by Equation (14);
    Fix $\mathbf{U}, \mathbf{R}, \mathbf{A}$ and $\mathbf{S}$, update $\mathbf{B}$ as illustrated in Step3;
    Fix $\mathbf{U}, \mathbf{B}, \mathbf{A}$ and $\mathbf{S}$, update $\mathbf{R}$ by optimizing:
        $\min_{\mathbf{R}} ||\mathbf{A} - \mathbf{RS}||_F^2 \quad s.t. ||\mathbf{R}_{\cdot i}||^2 \leq 1, \forall i$
    Fix $\mathbf{R}, \mathbf{B}, \mathbf{A}$ and $\mathbf{S}$, update $\mathbf{U}$ by optimizing:
        $\min_{\mathbf{U}} ||\mathbf{Y} - \mathbf{UA}||_F^2 \quad s.t. ||\mathbf{U}_{\cdot i}||^2 \leq 1, \forall i$
until convergency.
$\mathbf{H} = sign(\mathbf{A})$.
```

Consider the Lagrangian:

$$\mathcal{L}(\mathbf{B}, \vec{\theta}) = ||\mathbf{X} - \mathbf{BS}||_F^2 + \sum_{i=1}^{n} \theta_i (||\mathbf{B}_{\cdot i}|| - 1) \tag{16}$$

where $\theta_i > 0$ is the Lagrange multipliers. Letting the derivative of (16) with respect to $\mathbf{B}$ equal to zero, the close form solution of (15) can be obtained by

$$\mathbf{B} = \mathbf{XS}^T (\mathbf{SS}^T + \mathbf{\Theta})^{-1} \tag{17}$$

where $\mathbf{\Theta}$ is diagonal matrix whose diagonal entry $\mathbf{\Theta}_{ii} = \theta_i$, and is got by optimizing following Lagrange dual problem

$$\min_{\mathbf{\Theta}} Tr(\mathbf{XS}^T (\mathbf{SS}^T + \mathbf{\Theta})^{-1} \mathbf{SX}^T) + Tr(\mathbf{\Theta}) \quad s.t. \mathbf{\Theta}_{ii} \geq 0, \forall i \tag{18}$$

where $Tr(\cdot)$ denotes the trace of matrix, i.e. the sum of diagonal. Problem (18) can be solved by using Newtons method or conjugate gradient. The algorithm is summarized in Algorithm 1.

3.6 Extension to Out-of-Sample

In practice, the components of a new query can be quite diverse, now we discuss it in the following three situations.

Image only. We denote $\widetilde{\mathbf{x}}$ as original image feature of the query, and then obtain the sparse coding by solving

$$\min_{\mathbf{s}} ||\widetilde{\mathbf{x}} - \mathbf{Bs}||^2 + \lambda |\mathbf{s}| \tag{19}$$

where dictionary $\mathbf{B}$ is given by Algorithm 1. Let $\mathbf{s}^*$ be the optimal solution, then the hash codes $\mathbf{h} = sign(\mathbf{Rs}^*)$.

Text only. We denote $\widetilde{\mathbf{y}}$ as original text feature of the query, and the close form matrix factorization factor is

$$\mathbf{a}^* = (\mathbf{U}^T \mathbf{U})^{-1} (\mathbf{U}^T \widetilde{\mathbf{y}}) \tag{20}$$

In most cases, $\mathbf{U}$ is full column rank, then $(\mathbf{U}^T\mathbf{U})^{-1}$ exists. Otherwise, we may approximate $\mathbf{a}^*$ by $\widehat{\mathbf{a}}^* = (\mathbf{U}^T\mathbf{U} + \epsilon \mathbf{I})^{-1} (\mathbf{U}^T \widetilde{\mathbf{y}})$, where $\epsilon > 0$ is a small real number (e.g. 0.001), then we can get the hash codes $h = sign(\mathbf{a}^*)$.

Both Text and Image. We can use the same way to get hash codes described in **Image only** and **Text only**. Moreover, $\mathbf{a}^*$ also can be obtained according formula (14), which uses both image and text information, and then $h = sign(\mathbf{a}^*)$. We investigate the performance of cross-modal retrieval for these diverse queries in Section 4.2.6.

3.7 Discussion

In this section, we will show that LSSH is available for large-scale datesets. The time consuming for training LSSH includes sparse coding learning, latent semantic concepts learning, and Lagrange dual learning. Typically, solving (12) and (13) requires $O(nM^2)$ [2] and $O(d^3)$ respectively. The Lagrange dual (18), which is independent of sample size n, can be solved by using Newtons method or conjugate gradient, which has been shown more efficient than gradient descent [15]. In a word, the total time complexity of training LSSH is linear to n, which is really scalable for large-scale datesets compared with most existing cross-modal hashing methods.

4. EXPERIMENT

We conduct experiments on three real-world datasets for cross-modal similarity search to verify the effectiveness of LSSH. Specifically, datasets involved in our experiments consist of text and images, and we use text as query to search similar images and image as query to search similar texts. Furthermore, we analysis the parameter sensitivity, check the convergence property of Algorithm 1 and the influence of different query type to the similarity search performance.

4.1 Experiment Settings

4.1.1 Datasets

Wiki[3].The Wiki dataset was collected from Wikipedia consisting of 2,866 multimedia documents. Each document contains 1 image and at least 70 words. Each image is represented by a 128-dimension SIFT [18] histogram and each text is represented by a 10-dimension topics' vector generated by latent Dirichlet allocation (LDA) model [2]. Totally 10 categories are considered in this dataset and each document (image-text pair) is labeled by one of them. Documents are considered to be similar if they belong to the same category.

LabelMe[4]. The LabelMe dataset is created by MIT Computer Science and Artificial Intelligence Laboratory which is made up of 2688 images. Each image is annotated by several tags which denote the objects in this image, such as ”sea” and ”beach”. Tags occurs in less than 3 images are discarded and 245 unique tags remain. This dataset is divided to 8 unique outdoor scenes such as ”coast”, ”forest” and ”highway” and each image belongs to one scene. Each image is represented by a 512-dimension GIST [21] feature and each text is represented by an index vector of selected tags. Image-text pairs are regarded as similar if they share the same scene label.

NUS-WIDE[5]. The NUS-WIDE dataset is a real-word image dataset created by Lab for Media Search in National University of Singapore [28]. This dataset contains 81 concepts but some are scarce. So we select 10 most common

[2] The complexity of lasso algorithms is $O(nM^2 + M^3)$, but usually, $n \gg M$.

[3] http://www.svcl.ucsd.edu/projects/crossmodal/

[4] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[5] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

concepts, and thus 186,577 images are left from 269,648 images. Furthermore, we select 1000 most frequent tags from 5,018 unique tags in this dataset. Each image is represented by a 500-dimension SIFT histogram and each text is represented by an index vector of selected tags. Each image-text pair is annotated by at least 1 of 10 concepts. Pairs are considered to be similar if they share at least one concept.

4.1.2 Baseline Methods

Our method is compared against four state-of-the-art hashing methods for cross-modal similarity search as below.

- Cross-view Hashing[6] (CVH) [14] extends spectral hashing to the multi-view case and is a special case of IMH.
- Data Fusion Hashing[7](DFH) [3] constructs two groups of linear hash functions to preserve the similarity structure in each individual media type.
- Inter-media Hashing[6] (IMH) [27] introduces inter-media and intra-media consistency to discover a common hamming space, and uses linear regression with regularization model to learn view-specific hash functions.
- Composite Hashing with Multiple Information Sources[7] (CHMIS) [33] combines information from multi sources into integrated hash codes by optimizing the relaxed hash codes and combination coefficients alternatively.

CVH, IMH and DFH generates different hash codes for each modal of an instance, i.e., the hash codes of different modals of an instance is different, but these methods try to make these codes more similar for one instance. In our experiment, they will generate different hash codes for image and text separately of an instance(image-text pair). When a new image(text) query comes, they first generate its hash codes, and then search similar data from text(image) database. CHMIS generates unified hash codes for an instance combining all modals. However, if any one modal of an instance is unavailable, it can't generate hash codes for this instance, which is too demanding for real-world scenarios. This method improves search accuracy by combining multiple information sources of one instance, and actually is not implemented for cross-modal similarity search. Yet we still compare LSSH to CHMIS to verify the ability of LSSH to promote search performance by merging knowledge from heterogeneous data sources.

4.1.3 Evaluation Metrics

We adopt *mean Average Precision*(mAP) as the evaluation metric for effectiveness in our experiment. This evaluation metric has been widely used in literatures [27][35]. mAP has shown especially good discriminative power and stability to evaluate the performance of similarity search. A larger mAP indicates better performance that similar instances have high rank. Given a query and a set of R retrieved instances, the *Average Precision*(AP) is defined as

$$\mathrm{AP} = \frac{1}{L}\sum_{r=1}^{R} P(r)\delta(r)$$

where L is the number of relevant instances in retrieved set, $P(r)$ denotes the precision of top r retrieved instances which is defined as the ratio between the number of relevant instance and the number of retrieved instance r, and $\delta(r)$ is a indicator function which equals to 1 if the rth instance is relevant to query or 0 otherwise. Then the AP of all queries are averaged to obtain the mAP.

Furthermore, we also report two types of performance curves. One is *precision-recall* curve which shows the precision at different recall level. The other is *topN-precision* curve which reflects the change of precision with respect to the number of retrieved instances.

4.1.4 Implementation Details

The experiments are carried out as follows. For image data, we first apply Principle Component Analysis (PCA) to reduce the feature dimension to 64, which can also remove noise from image data. Then, the length of sparse codes, i.e., the size of dictionary B, is set to 512, and the sparsity parameter λ is set to 0.2. LSSH has two model parameters, μ which leverages the discrimination power between images and texts, and γ which controls the linear connection of latent semantic spaces. In the coming sections, we provide empirical analysis on parameter sensitivity, which verifies that LSSH can achieve stable and superior performance under a wide range of parameter values. When comparing with baseline methods, we use the following parameter settings for all experiments: $\mu = \gamma = 1$, which shows good performance on all three datasets. For baseline methods, we carefully tune the parameters for them and report the best results of them.

Furthermore, considering IMH and CHMIS requires too much resource to learn hash functions on NUS-WIDE with all data. We select randomly $10,000$ instances for all methods to train hash functions and then they are applied to the other instances in database to generate hash codes for them as in [27]. Moreover, hashing methods should have the ability to handle out-of-sample instances since data is keeping coming into database as time goes by in real world. So we simulate the situation by this experiment setting. For LSSH, hash codes for instances in database is generated as follows. We first generate sparse codes S for images by (7). Then we use (14) to generate unified codes A for all instances combining both images and texts. Finally, we apply sign function on A to obtain hash codes. And for query instances which only have one modal, i.e., image or text, we use methods introduced in 3.6 to generate hash codes. Moreover, we set $R = 100$, and all of the results are averaged over 10 runs. All the experiments are conducted on a computer which has Intel Xeon E5520 2.27GHz CPU, 16GB RAM.

4.2 Results and Discussions

4.2.1 Results on Wiki

We select 75% of the data as database and the rest as the query set. The mAP of LSSH and baseline methods are shown in Table 1., and the performance curves are shown in Figure 3. We can observe that LSSH can significantly outperform baseline methods on both cross-modal similarity search tasks which verifies the effectiveness of LSSH.

The semantic gap between two views of Wiki is quite large. In fact, text can better describe the topic of the image-text pair than image. When a image query comes, since it's not quite related to it's topic, it's difficult to find semantically similar texts. So the mAP of image query is low for all meth-

[6] We implemented it ourselves because the code is not publicly available.

[7] The source code is kindly provided by the authors.

Table 1: mAP Comparison on Three Datasets.

	Method	Wiki				LabelMe				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Img to Txt	CVH	0.1984	0.1490	0.1182	0.1133	0.4704	0.3694	0.2667	0.1915	0.4694	0.4656	0.4705	0.4777
	IMH	0.1922	0.1760	0.1572	0.1351	0.3593	0.2865	0.2414	0.1990	0.4564	0.4566	0.4589	0.4453
	DFH	0.2097	0.1995	0.1943	0.1898	0.4994	0.4213	0.3511	0.2788	0.4774	0.4677	0.4674	0.4703
	CHMIS	0.1942	0.1852	0.1796	0.1671	0.4894	0.4010	0.3414	0.2967	0.3596	0.3652	0.3565	0.3594
	LSSH	**0.2330**	**0.2340**	**0.2387**	**0.2340**	**0.6692**	**0.7109**	**0.7231**	**0.7333**	**0.4933**	**0.5006**	**0.5069**	**0.5084**
Txt to Img	CVH	0.2590	0.2042	0.1438	0.1170	0.5778	0.4403	0.3174	0.2153	0.4800	0.4688	0.4636	0.4709
	IMH	0.3717	0.3319	0.2877	0.2674	0.4346	0.3323	0.2771	0.2258	0.4600	0.4581	0.4653	0.4454
	DFH	0.2692	0.2575	0.2524	0.2540	0.5800	0.4310	0.3200	0.2313	0.5174	0.5077	0.4974	0.4903
	CHMIS	0.1942	0.1852	0.1796	0.1671	0.4894	0.4010	0.3414	0.2967	0.3596	0.3652	0.3565	0.3594
	LSSH	**0.5571**	**0.5743**	**0.5710**	**0.5577**	**0.6790**	**0.7004**	**0.7097**	**0.7140**	**0.6250**	**0.6578**	**0.6823**	**0.6913**

ods. Even so, LSSH can achieve best performance, especially with long hash codes. LSSH can reduce the semantic gap between modals in database which makes the hash codes of images quite related to the topics of instances. Actually, the hash codes for images are also for the instances. So when a text query which is highly related to its topic comes, it can obtain semantically similar images of it. But the hash codes of images generated by baseline methods still show little relevance to their topics. That's why LSSH can improve mAP by 18% at least which also shows the importance to reduce semantic gap between different modals.

We further observed that the PR-curve of several methods looks strange, e.g. the PR-curve of CVH for text query to image database at 64 bits shows that it behave like random guess in experiments. This phenomenon also happened in [35] and [36] and a reasonable explanation is given by [29]. Actually, all baseline methods are solved by eigenvalue decomposition and have orthogonality constraints on each bit so that each bit shows no correlation to each other. The first few projection directions may have high variance and their corresponding hash bits can be quite discriminative, which is quite useful to similarity search. However, as the code length increases, the hash codes will be dominated by bits with very low variance. Actually, since the variance is too low, the lower bits are meaningless and ambiguous. So these indiscriminative hash bits may lead the method to make random guess in experiments.

4.2.2 Results on LabelMe

75% of the data are chosen as the database and the remaining to form the query set. The mAP of LSSH and baseline methods are shown in Table 1. and Figure 4. shows the performance curves of them. LSSH shows more superior performance than baseline methods with different code length. Furthermore, the images and texts of LabelMe are quite related to each other, thus LSSH can learn more effective hash functions to increase similarity search performance by merging knowledge from heterogeneous data. Moreover, the mAP of image query is even higher than text query, this also shows the power of Sparse Coding to capture high-level semantic information of images.

We can also observe that as code length increase, LSSH performs better because LSSH can learn more precise descriptions for instances with more latent concepts and longer codes can encode more information. However, the performance of baseline methods degrades significantly as the increase of code length. This phenomenon has also been observed in [29] and [?]. The main reason is that the baseline methods are spectral-hashing-based methods which have orthogonality constraints on the projection directions to make each hash bits uncorrelated to each other. However, these orthogonality constraints sometimes lead to practical problem. It is well known that for most real-world datasets, most of the variance is contained in top few projections. With longer codes, these constraints will force them to progressively pick directions with low variance, which may reduce the quality and discriminative power of hash codes [29].

4.2.3 Results on NUS-WIDE

We select 2% of the data as the query set and the rest as the database. As mentioned above, we select $10,000$ instances from databases randomly as the training set to learn hash functions and then they are applied to other instances in database to generate hash codes. The mAP and performance curves are shown in Table 1. and Figure 5. respectively. Similar to results above, we observe that LSSH outperforms baseline methods significantly and it performs better with longer hash codes.

In real-world applications, the size of database can be so large that it's impossible to learn hash functions on the whole database because of the limitation of computational resources. And new data is keep coming into database as time goes by and hash codes for them need to be computed. So the hashing methods should be able to deal with out-of-sample instances (other instances in database and new coming instances). The common solution is to learn hash functions which project new data to a feature space and then quantize new features to binary hash codes by sign operation. The ability to deal with out-of-sample instances can test the effectiveness of hash functions which can further judge the ability of hashing methods to apply to practical problems. The experiment settings on NUS-WIDE is quite similar to real-world scenario. The experiment results show that LSSH can deal with out-of-sample instances easily and it has superior ability to handle large-scale database.

4.2.4 Parameter Sensitivity Analysis

We conduct empirical analysis on parameter sensitivity on all datasets, which validates that LSSH can achieve stable and superior performance under a wide range of parameter values, verifying that LSSH is robust to parameters.

When analyzing one parameter, we keep other parameters fixed to the settings mentioned in Section 4.1.4. Due to the limit of space, we only present the results at 64 bits on all datasets in Figure 6. The dashed lines are the best performance of baselines with all experiment settings, e.g. the red dashed line in the first figure shows the result of DFH at 16 bits, which, as be observed from Table 1., is the best result of all baselines varying code length for "Image to Text" task. We can observe that LSSH can outperform all best

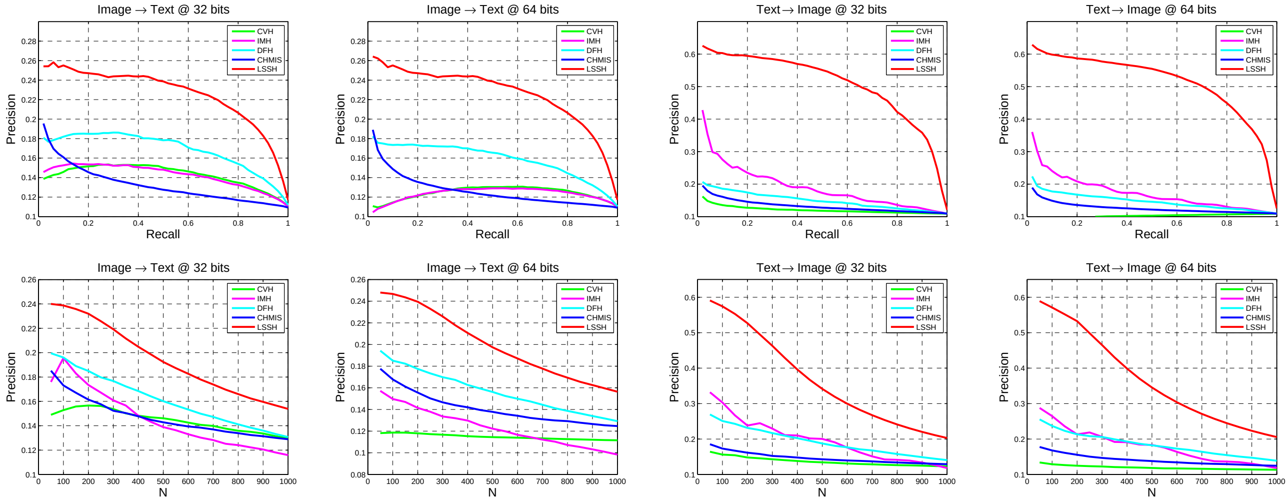

Figure 3: PR-Curves and topN-precision Curves on Wiki Varying Code Length

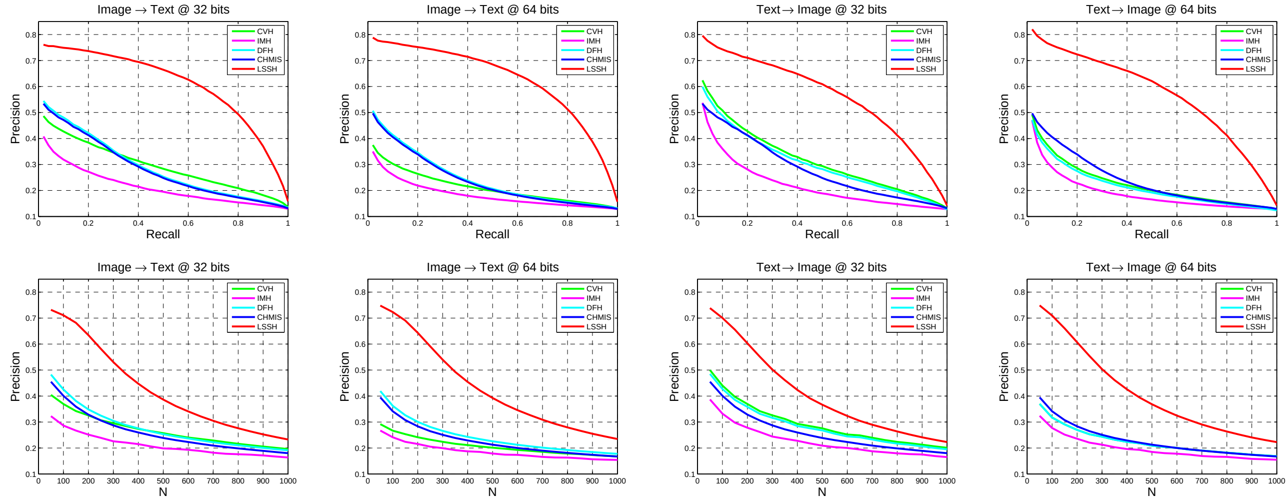

Figure 4: PR-Curves and topN-precision Curves on LabelMe Varying Code Length

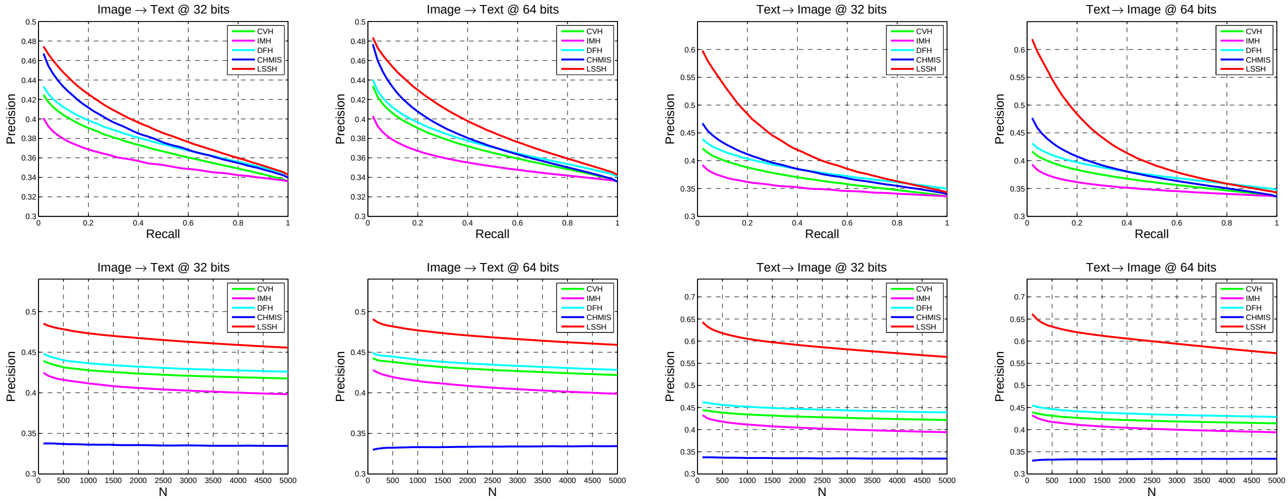

Figure 5: PR-Curves and topN-precision Curves on NUS-WIDE Varying Code Length

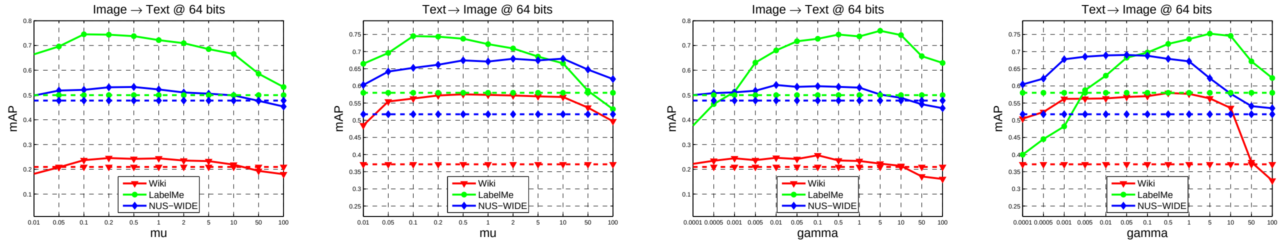

Figure 6: Parameter Sensitivity Analysis

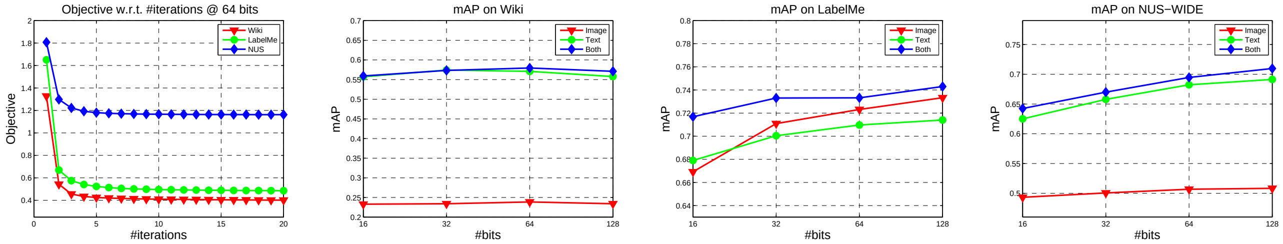

Figure 7: Convergence and Query Diversity Study

results of baselines when $\mu \in [0.05, 10]$ and $\gamma \in [0.005, 10]$.

The parameter μ leverages the power of images and texts. Actually, utilizing the information from both modals can lead to better results. When μ is too small, e.g., $\mu < 0.05$, our model just focuses on images while ignoring texts. When μ is too large, e.g., $\mu > 10$, our model prefers information from texts. Specifically, it's easy to choose a proper value for μ because we can observe that LSSH shows stable and superior performance when $\mu \in [0.05, 10]$.

The parameter γ controls the connection of latent semantic spaces. If γ is too small, the connection between different modals is weak with imprecise projection in formula (10), which will lead to poor performance for cross-modal similarity search. However, if γ is too large, the strong connection will make the learning of latent representations of images and texts, i.e., Sparse Coding and Matrix Factorization, to be quite imprecise. Because images and texts are represented by imprecise features, it's reasonable the the performance will degrade. Fortunately, it's also effortless to choose proper γ from the range $[0.005, 10]$;

4.2.5 *Convergence Study*

Since LSSH is solved by an iterative procedure, we empirically check its convergency property. Figure 7. shows that the value of objective function (averaged by the number of training data) can decrease steadily with more iterations and it can converge with 20 iterations on all datasets at 64 bits, which validates the effectiveness of Algorithm 1. The results at other code length are similar to at 64 bits.

Furthermore, the average time cost for each iteration at 64 bits on Wiki, LabelMe and NUS-WIDE is 3.98 seconds, 3.65 seconds and 12.73 seconds respectively. So we can see that LSSH can be solved with 10,000 training data in less than 5 minutes, which shows the high efficiency of Algorithm 1.

4.2.6 *Query Diversity Study*

As mentioned in Section 3.6, a coming query instance can be quite diverse, e.g., it may consists of a single image, or a single text or both. Here we test the influence of query type on the similarity search performance. When an image query comes, we use (7) to generate its codes while (20) is applied to text query. And as mentioned above, we use (7) and (14) to generate codes for instances with both image and text. Figure 7. shows the mAP results on three datasets of different query type varying code length from 16 to 128.

We can observe that query with both modals can slightly outperform the query with partial missing modal. This is reasonable because query with both modals contains more information and the hash codes for it can be more semantically informative, which leads to better performance.

Furthermore, the performance gap between image query and text query differs from datasets. In fact, for all datasets, texts belonging to one category show much similarity, so text query works well. The images in LabelMe belonging to one category, such as ”highway”, are quite similar to each other, that's why image query in LabelMe can achieve superior performance. But the images in Wiki are quite diverse. For example, images belonging to ”History” can be related to a building, a man or a weapon. So it's difficult to find semantically similar instances for an image query.

5. CONCLUSIONS

In this paper, we propose a novel hashing method, referred to as Latent Semantic Sparse Hashing, for large-scale cross-modal similarity search between images and texts. Specifically, we utilizes Sparse Coding to capture high level salient structures of images, and Matrix Factorization to extract latent concepts from texts. Then these high level semantic features are mapped to a joint abstraction space. The search performance can be promoted by merging multiple comprehensive latent semantic descriptions from heterogeneous data. We propose an iterative strategy which is highly efficient to explore the correlation between multi-modal representations and bridge the semantic gap between heterogeneous data in latent semantic space.

We conduct extensive experiments on three multi-modal datasets consisting of images and texts. Superior and stable performances of LSSH verifies the effectiveness of it compared against several state-of-the-art cross-modal hashing methods. With longer hash codes, LSSH can conduct matrix factorization more accurately and encode more information, which leads to better performance, while the baseline methods perform worse with longer hash codes because of the orthogonality constraints on their objective function. Experiments on NUS-WIDE, which is a large-scale datasets, show that LSSH can deal with out-of-sample easily and has the ability to handle large-scale database. The analysis on parameter sensitivity shows that LSSH is very robust to model parameters which can achieve stable and superior performance under a wide range of parameter values. Our convergence study shows the proposed learning algorithm is indeed effective and can be solved efficiently. And the study on query diversity shows the influence of different query types on the search performance and combining information from multiple source can help increase search performance.

6. ACKNOWLEDGMENTS

This research was supported by the National Basic Research Project of China (Grant No. 2011CB70700), the National HeGaoJi Key Project (No. 2013ZX01039-002-002), and the National Natural Science Foundation of China (Grant No. 61271394). And the authors would like to thank the reviewers for their valuable comments.

7. REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS'06*, pages 459–468, 2006.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

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

[4] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[5] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[6] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999.

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

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

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

[10] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STC*, pages 604–613. ACM, 1998.

[11] S. Kim, Y. Kang, and S. Choi. Sequential spectral learning to hash with multiple representations. In *ECCV*, pages 538–551. Springer, 2012.

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

[13] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, pages 2130–2137. IEEE, 2009.

[14] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, 2011.

[15] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[16] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE, 2012.

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

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[19] Z. Lu and Y. Peng. Latent semantic learning by efficient sparse coding with hypergraph regularization. In *AAAI*, 2011.

[20] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

[21] A. Oliva and T.Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001.

[22] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

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

[24] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia*, 9(5):923–938, 2007.

[25] R. Salakhutdinov and G. Hinton. Semantic hashing. *IJAR*, 50(7):969–978, 2009.

[26] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, pages 750–757. IEEE, 2003.

[27] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *ICMD*. ACM, 2013.

[28] R. H. H. L. Z. L. T. Chua, J. Tang and Y. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009.

[29] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.

[30] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. NIPS, 2008.

[31] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009.

[32] M. Yang, L. Zhang, J. Yang, and D. Zhang. Robust sparse coding for face recognition. In *CVPR*, 2011.

[33] D. Zhang, F. Wang, and L. Si. Composite hashing with multiple information sources. In *SIGIR*, 2011.

[34] D. Zhang, J. Wang, D. Cai, and J. Lu. Self-taught hashing for fast similarity search. In *SIGIR*, 2010.

[35] Y. Zhen and D. Yang. A probabilistic model for multimodal hash function learning. In *SIGKDD*, 2012.

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