Parametric Local Multimodal Hashing for Cross-view Similarity Search

Deming Zhai[1], Hong Chang[2], Yi Zhen[3], Xianming Liu[1], Xilin Chen[2], Wen Gao[1]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, China
[3]Hong Kong University of Science and Technology, Hong Kong, China
{dmzhai, hchang, xmliu, xlchen, wgao}@jdl.ac.cn, yzhen@cse.ust.hk

Abstract

Recent years have witnessed the growing popularity of hashing for efficient large-scale similarity search. It has been shown that the hashing quality could be boosted by hash function learning (HFL). In this paper, we study HFL in the context of multimodal data for cross-view similarity search. We present a novel multimodal HFL method, called Parametric Local Multimodal Hashing (PLMH), which learns a set of hash functions to locally adapt to the data structure of each modality. To balance locality and computational efficiency, the hashing projection matrix of each instance is parameterized, with guaranteed approximation error bound, as a linear combination of basis hashing projections of a small set of anchor points. A local optimal conjugate gradient algorithm is designed to learn the hash functions for each bit, and the overall hash codes are learned in a sequential manner to progressively minimize the bias. Experimental evaluations on cross-media retrieval tasks demonstrate that PLMH performs competitively against the state-of-the-art methods.

1 Introduction

The problem of similarity search is encouraged in many engineering and science applications, ranging from information retrieval, data mining, and computer vision. Many tasks such as content-based retrieval, object recognition and near-duplicate detection, depend critically on the choice of an effective and efficient similarity search strategy. To confront with the scalability issue in massive data, hashing-based methods have attracted considerable interest in recent years. The appealing property of hashing methods is they index data with binary hash codes which enjoy not only the compactness of the representation but also the low complexity in distance computation. By utilizing hash codes, we can achieve fast search with constant or sub-linear time complexity [Norouzi *et al.*, 2012] and greatly decreased storage cost.

Early exploration of hashing is data-independent [Andoni and Indyk, 2006; Lv *et al.*, 2007]. However, these methods require long codes to achieve good precision, which will result in low recall as the collision probability decreases exponentially with the code length. Recent endeavors aim at data-dependent hashing by employing machine learning techniques to learn the hash functions for specific datasets. This new direction is also referred to as hash function learning (HFL). Some representative HFL works include semantic hashing [Salakhutdinov and Hinton, 2007], spectral hashing [Weiss *et al.*, 2008], and kernel-based supervised hashing [Liu *et al.*, 2012], *etc*.

Existing HFL methods have been applied to a wide range of real-world applications with great success. Nevertheless, most algorithms deal with data lying in a single-view observation space, *i.e.*, unimodal hashing. Nowadays, it is common to conduct similarity search across different modalities. For instance, a multimedia search engine may perform queries in a corpus consisting of texts and images [Rasiwasia *et al.*, 2010; Chua *et al.*, 2009]. One may query image datasets by text keywords to quickly draw a vivid imagination, or query text datasets by images to accurately describe the details. Consequently, multimodal HFL, which learns different hash functions for different modalities to enable cross-modal comparison, is a very worthwhile direction to explore. This motivates the study reported in this paper.

Multimodal HFL is a challenging task, since different observations generally have different representations with incommensurable structure and dimensionality. Up to now, only a few attempts have been made for multimodal hashing [Zhen and Yeung, 2012a; Kumar and Udupa, 2011; Bronstein *et al.*, 2010; Zhen and Yeung, 2012b]. Bronstein *et al.* first put forward multimodal HFL method, called cross-modal similar sensitive hashing (CMSSH) [Bronstein *et al.*, 2010], which constructs two groups of linear hash functions (for bimodal case) sequentially based on a standard boosting framework. Later, Kumar *et al.* present another method called cross-view hashing (CVH) [Kumar and Udupa, 2011] which essentially extends spectral hashing to the multi-view setting. The main idea of CVH is to find view-specific linear projections as hash functions so that similar objects are mapped to similar binary codes across all of views. More recently, Zhen *et al.* present co-regularized hashing (CRH) [Zhen and Yeung, 2012a] for multimodal data based on a boosted co-regularization framework. The hash functions for each bit of hash codes are learned by solving DC (difference of convex functions) programs, while the learning for multiple bits is carried out via a boosting procedure.

The above multimodal HFL methods have achieved promising results in several applications, but they still have some limitations. First, these methods are all limited to a relatively narrow class of globally linear multimodal HFL that often cannot capture well the structure of the data for each modality. Although it is claimed that nonlinear hash functions could be achieved via the kernel trick [Schölkopf and Smola, 2001], the choice of both the problem-dependent kernel function and its parameters is still a sticky problem. Moreover, the nonlinearity is performed in a global way since the induced kernel function with the same parameters is generally applied to all data pairs in each view. Second, both CMSSH and CVH treat each bit independently, which results in finding unnecessary long hash codes. Besides, CMSSH ignores the intra-modality similarities which are very useful in some cases [Weiss *et al.*, 2008], while CRH does not consider the correlation within each modality in defining the intra-modality loss. A recent work, called multimodal latent binary embedding (MLBE) [Zhen and Yeung, 2012b], takes a probabilistic generative approach to solve the multimodal hashing problem. MLBE regards the binary latent factors as hash codes, and the learning of binary latent factors corresponds to hash function learning. MLBE alleviates the weaknesses of global linearity in each modality and bit independence in hash codes, and actually shows superior performance against most previous methods when the bit number is small. However, MLBE still has limitations on the restrictive global intra-modality weighting matrices involved in the probabilistic model. Moreover, since there are no explicit projections learned from multimodal inputs to presigned hash codes, MLBE suffers from the high computation complexity for the out-of-sample extension.

Actually, as pointed out by Bottou and Vapnik [Bottou and Vapnik, 1992], it is usually not easy to find a unique function which holds good predictability in the entire data space. Especially, very often a global hash function cannot accurately model the complex structure of large-scale datasets in which discriminative features vary from one neighborhood to the other. Vapnik [Vapnik, 1995] further demonstrate that local learning based algorithms usually achieve lower empirical errors than global ones. This is because nearby instances are more likely generated by the same data generation model, while far away instances tend to differ in it. Accordingly, neighboring instances may have the same or similar hash functions, while for instances lying away in different neighboring spaces their hash functions change heavily. Consequently, it is more ideal to learn a set of local multimodal hash functions which sufficiently boost the modeling ability.

In this paper, we propose a novel multimodal HFL method, called Parametric Local Multimodal Hashing (PLMH), which learns a set of local hash functions for each modality space. Different local hash functions are learned at different locations of the input spaces, therefore, the overall transformations of all points in each modality are locally linear but globally nonlinear. To reduce the computational complexity, the projection matrix of each instance is approximated as a linear combination of basis projections of a small set of anchor points, with guaranteed approximation error bound. For each bit of hash codes, we use the local optimal conjugate gradient method to optimize the relaxed objective function which preserves both intra-modality and inter-modality similarities. PLMH learns bits of hash codes in a sequential manner so that the bias introduced by antecedent hash functions can be sequentially minimized, and the dependence between bits is explored at the same time.

The rest of this paper is organized as follows. In Section 2, we present the PLMH method in detail, followed by a short discussion. Experiments analysis are presented in Section 3. Finally, Section 4 gives some conclusion remarks.

2 Parametric Local Multimodal Hash Function Learning

In this section, for simplicity of our presentation, we give a description on the bimodal case as an example, but it is very easy to extend the proposed method to more general cases including more than two modalities.

2.1 Local Hash Functions

Suppose that there are two sets of data points from different modalities: $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^{d_x}\}_{i=1}^{I}$ for a set of I images and $\mathcal{Y} = \{\mathbf{y}_j \in \mathbb{R}^{d_y}\}_{j=1}^{J}$ for a set of J textual documents. Each data point $\mathbf{x}_i(\mathbf{y}_j)$ is located in the $d_x(d_y)$ dimensional input feature space. We also have an inter-modality similarity matrix $\mathbf{S}^{xy} \in \{\pm 1, 0\}^{I\times J}$, where $+1$ and -1 denote similar and dissimilar relationships respectively, and 0 indicates the corresponding entries are not observed. In addition, two symmetric intra-modality similarity matrices $\mathbf{S}^x \in \mathbb{R}^{I\times I}$ and $\mathbf{S}^y \in \mathbb{R}^{J\times J}$ are constructed to reflect the local geometric structures of data in each input feature space.

We solve the problem of generating K-bit hash codes through parametric local multimodal hash functions learning. Specifically, two sets of local hash functions are defined as:

$$f(\mathbf{x}_i) = \mathrm{sgn}(\mathbf{W}_i^T\mathbf{x}_i) \text{ and } \mathrm{g}(\mathbf{y}_j) = \mathrm{sgn}(\mathbf{W}_j^T\mathbf{y}_j), \quad (1)$$

where $\mathrm{sgn}(\cdot)$ denotes the element-wise sign function, $\mathbf{W}_i \in \mathbb{R}^{d_x\times K}$ and $\mathbf{W}_j \in \mathbb{R}^{d_y\times K}$ are the instance-specific projection matrices. Note that the projection matrix is defined for each individual data point but not shared by all data points globally in each view. The overall transformations of all points are locally linear but globally nonlinear. As a result, different local hash functions are learned at different locations of the input space for each modality.

To balance locality and computational efficiency, we propose to approximate the projection matrix for each point instead of directly learning it. Specially, inspired by local coordinate coding (LCC) [Yu *et al.*, 2009], we parameterize the local projection matrix $\mathbf{W}_i$ as a linear weighted combination of a small set of projection basis associated with a set of anchor points:

$$\mathbf{W}_i = \sum_{\mathbf{x}_{a_k}\in\mathcal{C}_x} \gamma_{a_k}(\mathbf{x}_i)\mathbf{W}_{a_k}. \quad (2)$$

Here, $\mathcal{C}_x = \{\mathbf{x}_{a_1}, \ldots, \mathbf{x}_{a_m}\}$ represents the set of anchor points, where $\{a_k\}_{k=1}^{m}$ are the indices of the anchor points in $\mathcal{X}$, and $m = |\mathcal{C}_x|$. The vector of combination weights is denoted as $\boldsymbol{\gamma}(\mathbf{x}_i) = [\gamma_{a_1}(\mathbf{x}_i), \ldots, \gamma_{a_m}(\mathbf{x}_i)]^T$. That is,

the weights are decided locally by the anchor points. Both $\mathcal{C}_x$ and $\boldsymbol{\gamma}(\cdot)$ are unknown parameters to be learned by our algorithm. Similarly, we can represent the set of anchor points and the combination weights for the projection matrix of each data point in $\mathcal{Y}$ as $\mathcal{C}_y = \{\mathbf{y}_{b_1}, \ldots, \mathbf{y}_{b_m}\}$ and $\boldsymbol{\theta}(\mathbf{y}_j) = [\theta_{b_1}(\mathbf{y}_j), \ldots, \theta_{b_m}(\mathbf{y}_j)]^T$.

The feasibility and error bound of the above parameterizations can be verified. To this end, we rewrite $\mathbf{W}_i$ as a vector-valued function: $h(\mathbf{x}_i) = \mathrm{vec}(\mathbf{W}_i)$, where $\mathrm{vec}(\cdot)$ denotes the operator to convert a matrix into a vector in a column-wise manner. It can be proved that $h(\cdot)$ is a Lipschitz smooth function according to the following definition.

Definition 2.1 ***Lipschitz Smooth Vector-Valued Function*** *[Wang* et al.*, 2012a]: A vector-valued function $h(\mathbf{x})$ on $\mathbb{R}^{d_x}$ is a (α, β, p)-Lipschitz smooth function with respect to a vector norm $||\cdot||$, if $||h(\mathbf{x}) - h(\mathbf{x}')|| \leq \alpha||\mathbf{x} - \mathbf{x}'||$ and $||h(\mathbf{x}) - h(\mathbf{x}') - \nabla h(\mathbf{x}')^T(\mathbf{x} - \mathbf{x}')|| \leq \beta||\mathbf{x} - \mathbf{x}'||^{1+p}$, where $\nabla h(\mathbf{x}')$ is the derivative of the function h at $\mathbf{x}'$, $\alpha, \beta > 0$ and $p \in (0, 1]$.*

It has been shown in [Yu *et al.*, 2009] and [Wang *et al.*, 2012a] that any Lipshitz smooth real-valued and vector-valued functions can be approximated as a linear combination of the function values on the anchor points. Based on these results, we have the following lemma that gives the error bound for Eqn. (2).

Lemma 2.1 ***Error Bound for Approximated Local Hashing Projection****: Suppose the projection matrix $\mathbf{W}_i$ is approximated by a set of local projection basis as in (2), parameterized by $\mathcal{C}_x$ and $\boldsymbol{\gamma}(\mathbf{x}_i)$. We have the following relationship for all $\mathbf{x}_i \in \mathcal{X}$:*

$$\left\|\mathbf{W}_i - \sum_{\mathbf{x}_{a_k} \in \mathcal{C}} \gamma_{a_k}(\mathbf{x}_i)\mathbf{W}_{a_k}\right\| \leq \alpha \left\|\mathbf{x}_i - \sum_{\mathbf{x}_{a_k} \in \mathcal{C}} \gamma_{a_k}(\mathbf{x}_i)\mathbf{x}_{a_k}\right\| + \beta \sum_{\mathbf{x}_{a_k} \in \mathcal{C}} \gamma_{a_k}(\mathbf{x}_i) \left\|\mathbf{x}_i - \mathbf{x}_{a_k}\right\|^{1+p}. \quad (3)$$

Due to the space limitation, the proof of the above Lemma is omitted here.

Based on the verified local parameterization defined above, the local hash functions could be further formulated with respect to the projection basis as follows:

$$f(\mathbf{x}_i) = \mathrm{sgn}\Big(\sum_{\mathbf{x}_{a_k} \in \mathcal{C}} \gamma_{a_k}(\mathbf{x}_i)\mathbf{W}_{a_k}^T\mathbf{x}_i\Big) = \mathrm{sgn}\left(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_i) \otimes \mathbf{I})\mathbf{x}_i\right)$$
$$\text{and } g(\mathbf{y}_j) = \mathrm{sgn}\left(\mathbf{W}_y^T(\boldsymbol{\theta}(\mathbf{y}_j) \otimes \mathbf{I})\mathbf{y}_j\right), \quad (4)$$

where $\mathbf{W}_x = [\mathbf{W}_{a_1}; \cdots; \mathbf{W}_{a_m}]$ and $\mathbf{W}_y = [\mathbf{W}_{b_1}; \cdots; \mathbf{W}_{b_m}]$ are the concatenated basis projection matrices, $\mathbf{I}$ is the identity matrix, and $\otimes$ represents the kronecker product. This transformation can be regarded as an explicit kernel projection of data in a $d_x(d_y)$-dimensional space to a higher $md_x(md_y)$-dimensional space expanded by anchor points.

To obtain the anchor points and the weights for all instances, a straightforward manner is to formulate it as an optimization problem by minimizing the reconstruction error bound in Eqn. (3) as that in LCC method [Yu *et al.*, 2009]. However, solving such an optimization problem is computationally expensive. Alternatively, we adopt a much more efficient and practical method in our experiments. We simply set the anchor points as the outputs of k-median (setting $k = m$) algorithm and compute the corresponding weights using inverse Euclidian distance [Gemert *et al.*, 2008] based weighting for nearest neighbors. Empirical results show that this does not lead to a decrease of the performance compared with minimizing codings [Yu *et al.*, 2009].

2.2 Objective Function

In devising the multimodal HFL algorithm, two important issues should be taken into consideration: (1) The inter-modality pairs with the same (or similar) semantic concept should be mapped to the same hash bin, and vice versa. (2) The intra-modality local topological structures should be preserved, which forces points with high similarity to have similar binary codes in the hamming space for each modality.

We propose to achieve the overall objective by minimizing the following energy function *w.r.t.* $\{\mathbf{W}_{a_k}\}_{k=1}^m$ and $\{\mathbf{W}_{b_k}\}_{k=1}^m$:

$$\mathcal{O} = \sum_{i=1}^{I}\sum_{j=1}^{J} \ell_{ij}^{xy} + \frac{\mu}{2}\left(\sum_{i=1}^{I}\sum_{i'=1}^{I} \ell_{ii'}^{x} + \sum_{j=1}^{J}\sum_{j'=1}^{J} \ell_{jj'}^{y}\right), \quad (5)$$

where ℓ_{ij}^{xy} is the pairwise inter-modality loss term, $\ell_{ii'}^{x}$ and $\ell_{jj'}^{y}$ are the pairwise intra-modality loss terms for modalities $\mathcal{X}$ and $\mathcal{Y}$, respectively. μ is the regularization parameter to balance these two kinds of loss terms. In this work, we define inter-modality loss term as:

$$\ell_{ij}^{xy} = ||\mathrm{sgn}(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_i) \otimes \mathbf{I})\mathbf{x}_i) - \mathrm{sgn}(\mathbf{W}_y^T(\boldsymbol{\theta}(\mathbf{y}_j) \otimes \mathbf{I})\mathbf{y}_j)||^2 \mathbf{S}_{ij}^{xy},$$

where $\mathbf{S}^{xy}$ represents the inter-modality similarity matrix. Inspired by Laplacian eigenmap [Belkin and Niyogi, 2003], the pairwise intra-modality loss terms $\ell_{ii'}^{x}$ and $\ell_{jj'}^{y}$ are defined as:

$$\ell_{ii'}^{x} = ||\mathrm{sgn}(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_i) \otimes \mathbf{I})\mathbf{x}_i) - \mathrm{sgn}(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_{i'}) \otimes \mathbf{I})\mathbf{x}_{i'})||^2 \mathbf{S}_{ii'}^{x}$$
$$\ell_{jj'}^{y} = ||\mathrm{sgn}(\mathbf{W}_y^T(\boldsymbol{\theta}(\mathbf{y}_j) \otimes \mathbf{I})\mathbf{y}_j) - \mathrm{sgn}(\mathbf{W}_y^T(\boldsymbol{\theta}(\mathbf{y}_{j'}) \otimes \mathbf{I})\mathbf{y}_{j'})||^2 \mathbf{S}_{jj'}^{y},$$

where $\mathbf{S}^x$ and $\mathbf{S}^y$ are the intra-modality similarity matrices.

Note that the inner product of the hashing code of each instance always equals to the constant hash number, meaning:

$$\left(\mathrm{sgn}\left(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_i) \otimes \mathbf{I})\mathbf{x}_i\right)\right)^T \mathrm{sgn}\left(\mathbf{W}_x^T(\boldsymbol{\gamma}(\mathbf{x}_i) \otimes \mathbf{I})\mathbf{x}_i\right) = K.$$

By neglecting the constant term, the minimization problem in (5) equals to maximizing the following objective function:

$$\mathcal{J} = 2\mathrm{Tr}\left(\mathrm{sgn}(\mathbf{W}_x^T\mathbf{X})\mathbf{S}^{xy}\mathrm{sgn}(\mathbf{Y}^T\mathbf{W}_y)\right) + \mu\mathrm{Tr}\left(\mathrm{sgn}(\mathbf{W}_x^T\mathbf{X})\mathbf{S}^{x}\mathrm{sgn}(\mathbf{X}^T\mathbf{W}_x) + \mathrm{sgn}(\mathbf{W}_y^T\mathbf{Y})\mathbf{S}^{y}\mathrm{sgn}(\mathbf{Y}^T\mathbf{W}_y)\right), \quad (6)$$

where $\mathbf{X} = [(\boldsymbol{\gamma}(\mathbf{x}_1) \otimes \mathbf{I})\mathbf{x}_1, \cdots, (\boldsymbol{\gamma}(\mathbf{x}_I) \otimes \mathbf{I})\mathbf{x}_I]$, and $\mathbf{Y} = [(\boldsymbol{\theta}(\mathbf{y}_1) \otimes \mathbf{I})\mathbf{y}_1, \cdots, (\boldsymbol{\theta}(\mathbf{y}_J) \otimes \mathbf{I})\mathbf{y}_J]$.

2.3 Optimization Solutions

Maximizing Eqn. (6) is a non-trivial problem since the objective function is neither convex nor smooth. It is observed that the same sign is more favored for similar pairs and different signs are more favored for dissimilar pairs, regardless of their

magnitudes. As a result, $\mathcal{J}$ can be approximated by replacing the sign of projections with the signed magnitude:

$$\begin{aligned}\mathcal{J} =& 2\mathrm{Tr}\left(\mathbf{W}_x^T\mathbf{X}\mathbf{S}^{xy}\mathbf{Y}^T\mathbf{W}_y\right) + \\ & \mu\mathrm{Tr}\left(\mathbf{W}_x^T\mathbf{X}\mathbf{S}^x\mathbf{X}^T\mathbf{W}_x + \mathbf{W}_y^T\mathbf{Y}\mathbf{S}^y\mathbf{Y}^T\mathbf{W}_y\right). \quad (7)\end{aligned}$$

Finally, we impose an additional constraint to achieve the scaling invariance and bits decorrelation, and solve the optimization problem of local hash functions by maximizing:

$$\mathcal{J} = \mathrm{Tr}\left(\mathbf{W}^T\mathbf{Z}\mathbf{S}\mathbf{Z}^T\mathbf{W}\right) \ s.t.\ \mathbf{W}^T\mathbf{Z}\mathbf{Z}^T\mathbf{W} = \mathbf{I}, \quad (8)$$

where $\mathbf{W} = \begin{bmatrix} \mathbf{W}_x \\ \mathbf{W}_y \end{bmatrix}$, $\mathbf{Z} = \begin{bmatrix} \mathbf{X} & \\ & \mathbf{Y} \end{bmatrix}$, and $\mathbf{S} = \begin{bmatrix} \mu\mathbf{S}^x & \mathbf{S}^{xy} \\ (\mathbf{S}^{xy})^T & \mu\mathbf{S}^y \end{bmatrix}$.

Although the optimal solutions can be derived directly by solving an eigen-decomposition problem $\mathbf{Z}\mathbf{S}\mathbf{Z}^T\mathbf{W} = \lambda\mathbf{Z}\mathbf{Z}^T\mathbf{W}$ in theory, it is rather time consuming and storage prohibitive to optimize all parameters simultaneously in practice. To be more efficient, we propose to learn the hash functions for each bit of hash codes via the local optimal conjugate gradient method. Rewrite $\mathbf{W}_x$ and $\mathbf{W}_y$ as $\mathbf{W}_x = [\mathbf{w}_x^1, \cdots, \mathbf{w}_x^K]$ and $\mathbf{W}_y = [\mathbf{w}_y^1, \cdots, \mathbf{w}_y^K]$. Let $\mathbf{Z}^{k-1}$ be the updated data matrix after removing the spanned subspace generated from previous k-1 projection directions. The details of solving one-bit multimodal hash functions are described in Algorithm 1.

Algorithm 1 One-bit multimodal HFL via conjugate gradient

Input: $\mathbf{S}$, $\mathbf{Z}$, $\mathbf{Z}^{k-1}$ and the current bit $k \in \{1, \cdots, K\}$
Output: $\mathbf{w}_x^k$ and $\mathbf{w}_y^k$
Procedure:
1. Initialize $\mathbf{w}_x^{k(0)}$ and $\mathbf{w}_y^{k(0)}$, $\mathbf{w}^{k(0)} = [\mathbf{w}_x^{k(0)}; \mathbf{w}_y^{k(0)}]$;
   Let $\mathbf{E} = \mathbf{Z}\mathbf{S}\mathbf{Z}^T$, $\mathbf{F} = \mathbf{Z}^{k-1}(\mathbf{Z}^{k-1})^T$;
2. For $i = 0, \cdots$, until convergence
   2.1 Compute the ratio $\rho(\mathbf{w}^{k(i)}) = \frac{(\mathbf{w}^{k(i)})^T\mathbf{E}\mathbf{w}^{k(i)}}{(\mathbf{w}^{k(i)})^T\mathbf{F}\mathbf{w}^{k(i)}}$;
   2.2 Compute the conjugate gradient:
   $r(\mathbf{w}^{k(i)}) = \mathbf{E}\mathbf{w}^{k(i)} - \rho(\mathbf{w}^{k(i)})\mathbf{F}\mathbf{w}^{k(i)}$;
   2.3 Search the step size ζ^i via Rayleigh-Ritz method;
   2.4 Update $\mathbf{w}^{k(i+1)} := \mathbf{w}^{k(i)} + \zeta^i r(\mathbf{w}^{k(i)})$.
End for

To learn the hash functions for multiple bits, we propose to generalize the sequential learning process (SLP) [Wang *et al.*, 2012b] for local multimodal HFL to progressively minimize the bias introduced by antecedent hash functions. The overall procedure of PLMH method is represented in Algorithm 2. At each iteration, both the intra and inter similarities are updated (Step 3.2-3.3) by imposing higher weights on point pairs violated by previous hash functions (*i.e.*, the condition in Step 3.3), and the data matrices in constraint term are also updated (Step 3.4-3.5) to decorrelate the hash bits for compact codes. Since the original data matrices are still exploited in the objective function term (to compute matrix $\mathbf{E}$ in Algorithm 1), SLP tends to progressively pick the hash projections with high empirical accuracy as well as low correlation on data sets. If the empirical accuracy and bits decorrelation contradict each other, the algorithm trades decorrelation for accuracy by picking correlated projections. Consequently, the sequential process also implicitly creates the dependency between bits.

Algorithm 2 Parametric Local Multimodal Hashing (PLMH)

Input:
$\mathcal{X}$, $\mathcal{Y}$: multimodal data
$\mathbf{S}^{xy}$, $\mathbf{S}^{x(y)}$: inter-modality and intra-modality similarity
μ: regularization parameter
K: code length
Output:
$\mathbf{w}_x^k = [\mathbf{w}_{a_1}^k; \cdots; \mathbf{w}_{a_m}^k]$, $k = 1, \ldots K$
$\mathbf{w}_y^k = [\mathbf{w}_{b_1}^k; \cdots; \mathbf{w}_{b_m}^k]$, $k = 1, \ldots K$
Procedure:
1. $(\mathcal{C}_x, \{\boldsymbol{\gamma}(\mathbf{x}_i)\}_{i=1}^I)$ = LocalCoding$\left(\{\mathbf{x}_i\}_{i=1}^I\right)$;
   $(\mathcal{C}_y, \{\boldsymbol{\theta}(\mathbf{y}_j)\}_{j=1}^J)$ = LocalCoding$\left(\{\mathbf{y}_j\}_{j=1}^J\right)$;
2. Compute $\mathbf{X}$ and $\mathbf{Y}$ in Eqn. (6); set $\mathbf{Z}^0 = \mathbf{Z}$;
3. For k = 1 to K do
   3.1 Optimize $\mathbf{w}_x^k$ and $\mathbf{w}_y^k$ via Algorithm 1;
   3.2 Compute the similarity of current hash projections:
   $\widetilde{\mathbf{S}}^x = \mathbf{X}^T\mathbf{w}_x^k(\mathbf{w}_x^k)^T\mathbf{X}$, $\widetilde{\mathbf{S}}^y = \mathbf{Y}^T\mathbf{w}_y^k(\mathbf{w}_y^k)^T\mathbf{Y}$,
   and $\widetilde{\mathbf{S}}^{xy} = \mathbf{X}^T\mathbf{w}_x^k(\mathbf{w}_y^k)^T\mathbf{Y}$;
   3.3 Update the similarity matrices:
   $\mathbf{S}^r = \mathbf{S}^r - T(\widetilde{\mathbf{S}}^r, \mathbf{S}^r), r \in \{x, y, xy\}$,
   $s.t.\ T(\widetilde{\mathbf{S}}_{ij}^r, \mathbf{S}_{ij}^r) = \begin{cases} \widetilde{\mathbf{S}}_{ij}^r & \mathrm{sgn}(\mathbf{S}_{ij}^r \cdot \widetilde{\mathbf{S}}_{ij}^r) < 0 \\ 0 & otherwise \end{cases}$;
   3.4 Compute the residual of data matrix:
   $\mathbf{X}^k = \mathbf{X}^{k-1} - \mathbf{w}_x^k(\mathbf{w}_x^k)^T\mathbf{X}^{k-1}$, $\mathbf{Y}^k$ (similar as $\mathbf{X}^k$);
   3.5 Update $\mathbf{S} = \begin{bmatrix} \mu\mathbf{S}^x & \mathbf{S}^{xy} \\ (\mathbf{S}^{xy})^T & \mu\mathbf{S}^y \end{bmatrix}$ and $\mathbf{Z}^k = \begin{bmatrix} \mathbf{X}^k & \\ & \mathbf{Y}^k \end{bmatrix}$.
End for

2.4 Complexity Analysis and Discussions

The computational cost of the proposed algorithm is mainly on three modules: local coding (Step 1-2), one-bit multimodal HFL (Step 3.1), and sequential learning update (Step 3.2-3.5). For the stage of local coding, the time complexity is linear with the number of data points. For the stage of one-bit multimodal HFL, the computationally expensive part is on evaluating the conjugate gradient, with the time complexity $O(D_x^2 N_{S^x} + D_y^2 N_{S^y} + D_x D_y N_{S^{xy}} + D_x^2 I + D_y^2 J)$, where $D_{x(y)} = md_{x(y)}$, $N_{S^{x(y)}}$ and $N_{S^{xy}}$ are the number of observations in the intra-modality and inter-modality similarities, respectively. Since in general the similarity matrices are sparse, the complexity for conjugate gradient can be greatly reduced. For the stage of sequential learning update, the complexity is $O(D_x I + D_y J + N_{S^x} + N_{S^y} + N_{S^{xy}})$. To summarize, the time complexity of our algorithm scales linearly with the number of data points and is quadratic with the data dimension. Thus the proposed algorithm is computationally efficient as long as the dimensions of input features are not exceptionally large.

It is worth mentioning that our work is also closely related to two recent works, locally linear support vector ma-

chines (LLSVM) [Ladicky and Torr, 2011] and parametric local metric learning (PLML) [Wang *et al.*, 2012a], which share the same idea of local learning. However, there are several significant differences between the proposed method and them. First, our method aims to do multimodal hashing, while LLSVM is developed for classification and PLML is for continuous Mahalanobis metric learning. Second, LLSVM and PLML are designed for dealing with data in a single view observation space, which is quite different from the multi-view scenarios of interest here. Third and more importantly, the strategy of local learning proposed in our method is totally different from that of LLSVM and PLML.

3 Experiments

Empirical studies are conducted on cross-media retrieval application for two typical tasks: (1) querying image database by text keywords; (2) query text database by image examples.

We evaluate the performance of the proposed PLMH method, and compare it with four state-of-the-art multimodal HFL methods: (1) CMSSH [Bronstein *et al.*, 2010]; (2) CVH [Kumar and Udupa, 2011]; (3) CRH [Zhen and Yeung, 2012a]; (4) MLBE [Zhen and Yeung, 2012b]. [1]

The retrieval performance is evaluated by mean Average Precision (mAP). For each query and a set of R retrieved documents, we first compute the Average Precision (AP). We then average the AP values over all the queries in the query set to obtain the mAP measure. The larger the mAP, the better the retrieval performance. In the experiments, we set $R = 50$. Moreover, we also report the precision-recall curve and recall curve by varying the Hamming radius of the retrieved points.

For PLMH, the intra-modality similarity matrices are computed as $s = e^{-d^2/2\sigma^2}$ within 8 nearest neighbors (8NN), where d is the Euclidean distance between two normalized feature vectors, and σ^2 is fixed to 1 for both data sets. The inter-modality similarity matrices are simply determined by the class labels with 0.1% randomly selected entries observed. Besides, 20 and 50 anchors points of each modality are generated with k-median clustering for flickr and wiki datasets, respectively[2]. The weights of local coding are simply obtained using inverse Euclidian distance [Gemert *et al.*, 2008] based weighting solved for 8NN. The regularization parameter μ is set to 1.

In experiments, 3000 instances are randomly selected as the training set from the database for each modality. All comparison methods adopt the same training data for fairness.

3.1 Results on Wiki

The Wikipedia dataset [Rasiwasia *et al.*, 2010] from the Wikipedia's "featured articles", consists of 2866 documents which are image-text pairs and annotated with labels of 10 semantic categories. Each image is represented *w.r.t.* a 128-dimensional SIFT [Lowe, 2004] codebook and each text is represented with a 10-dimensional latent dirichlet allocation model [Blei *et al.*, 2003]. We use 80% of the data as the database and the remaining 20% to form the query set.

[1]The codes of these methods are provided by the authors.

[2]More complex problems will often require a larger number of anchor points to better model the complex structure of the data.

The mAP results are summarized in Table 1 with various code lengths. The precision-recall curves and recall curves are illustrated in Figrue 1[3]. From the results, we can see that PLMH outperforms all comparative studies under all settings. Benefiting from the local hash functions, PLMH method achieves higher empirical accuracy than the global based ones.

Table 1: mAP comparison on Wiki

Task	Method	Code Length		
		$K = 8$	$K = 16$	$K = 24$
Image Query v.s. Text Database	CMSSH	0.1988	0.1973	0.1785
	CVH	0.2758	0.2134	0.1843
	CRH	0.2956	0.2714	0.2607
	MLBE	0.3175	0.2507	0.2736
	PLMH	**0.4435**	**0.4916**	**0.5516**
Text Query v.s. Image Database	CMSSH	0.1912	0.2128	0.1977
	CVH	0.3342	0.2893	0.2839
	CRH	0.3260	0.3222	0.3407
	MLBE	0.4414	0.2977	0.2869
	PLMH	**0.5976**	**0.6073**	**0.6011**

3.2 Results on Flickr

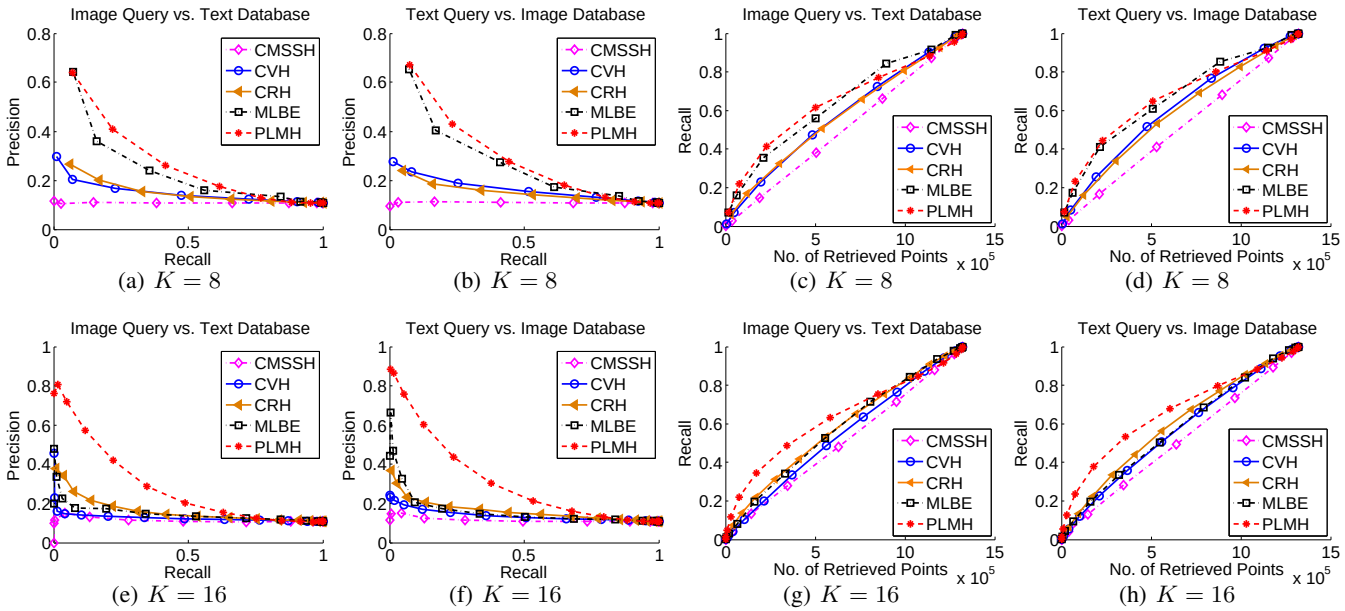
The Flickr dataset contains 186,577 image-tag pairs from the 10 largest classes of NUS dataset [Chua *et al.*, 2009]. Each image is represented with a 500-dimensional SIFT codebook and each tag is represented with a 1000-dimensional feature vector which is reduced by PCA on original tag occurrence features. We use 99% of the data as the database and the remaining 1% to form the query set.

Table 2: mAP comparison on Flickr

Task	Method	Code Length		
		$K = 8$	$K = 16$	$K = 24$
Image Query v.s. Text Database	CMSSH	0.4612	0.4724	0.5103
	CVH	0.5287	0.5000	0.4461
	CRH	0.5385	0.5290	0.5393
	MLBE	0.5394	0.5235	0.4805
	PLMH	**0.5987**	**0.5976**	**0.5975**
Text Query v.s. Image Database	CMSSH	0.5277	0.5133	0.4588
	CVH	0.5259	0.4949	0.4491
	CRH	0.5225	0.5217	0.5244
	MLBE	0.5542	0.5293	0.4852
	PLMH	**0.5701**	**0.5947**	**0.5985**

Similar to previous subsection, we show the mAP results in Table 2, the precision-recall curves and recall curves in Figure 2. We can find that PLMH achieves the best overall performance. It further verifies that the proposed local hash functions learning is more robust and can better model the complex structure of large-scale datasets compared with state-of-the-art methods.

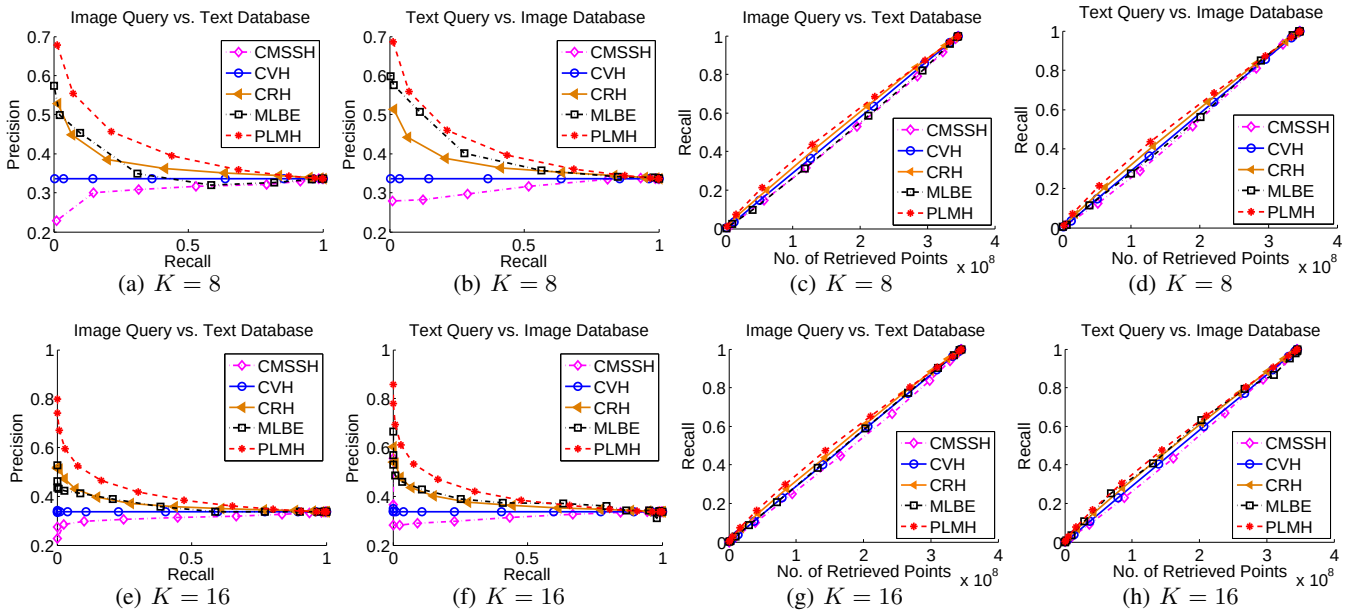
[3]Due to space limitation, only $K = 8, 16$ are shown in Figrue 1.



(a) $K = 8$ (b) $K = 8$ (c) $K = 8$ (d) $K = 8$

(e) $K = 16$ (f) $K = 16$ (g) $K = 16$ (h) $K = 16$

Figure 1: Precision-recall curves and recall curves on Wiki



(a) $K = 8$ (b) $K = 8$ (c) $K = 8$ (d) $K = 8$

(e) $K = 16$ (f) $K = 16$ (g) $K = 16$ (h) $K = 16$

Figure 2: Precision-recall curves and recall curves on Flickr

4 Conclusions

In this paper, we presented a new parametric local multimodal hashing (PLMH) method for cross-view similarity search. Different local hash functions are learned at different locations of the input spaces to better model the complex structure of large-scale datasets. As a result, PLMH achieves higher empirical query accuracy than global-based ones. Comparatives studies on two benchmark datasets show that PLMH outperforms the state-of-the-art multimodal hashing methods. In future work, we will explore more efficient optimization algorithms to further improve the scalability of PLMH.

Acknowledgement: This work was supported by the Major State Basic Research Development Program of China (973 Program) under Grant 2009CB320900, the National Science Foundations of China under Grant 61272319, and the new Ph.D researcher award of Chinese Ministry of Education.

References

[Andoni and Indyk, 2006] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, 2006.

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[Bottou and Vapnik, 1992] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

[Bronstein *et al.*, 2010] M.M. Bronstein, A.M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 48:1–48:9, New York, NY, USA, 2009. ACM.

[Gemert *et al.*, 2008] Jan C. Gemert, Jan-Mark Geusebroek, Cor J. Veenman, and Arnold W. Smeulders. Kernel codebooks for scene categorization. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 696–709, Berlin, Heidelberg, 2008. Springer-Verlag.

[Kumar and Udupa, 2011] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, IJCAI, 2011.

[Ladicky and Torr, 2011] Lubor Ladicky and Philip H. S. Torr. Locally linear support vector machines. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[Lowe, 2004] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2), November 2004.

[Lv *et al.*, 2007] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, 2007.

[Norouzi *et al.*, 2012] M. Norouzi, A. Punjani, and D.J. Fleet. Fast search in hamming space with multi-index hashing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[Rasiwasia *et al.*, 2010] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the international conference on Multimedia*, MM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[Salakhutdinov and Hinton, 2007] R. Salakhutdinov and G. Hinton. Semantic hashing. In *SIGIR workshop on Information Retrieval and applications of Graphical Models*, 2007.

[Schölkopf and Smola, 2001] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT, 2001.

[Vapnik, 1995] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[Wang *et al.*, 2012a] Jun Wang, Alexandros Kalousis, and Adam Woznica. Parametric local metric learning for nearest neighbor classification. In *Advances in Neural Information Processing Systems 25*, pages 1610–1618. 2012.

[Wang *et al.*, 2012b] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2393–2406, 2012.

[Weiss *et al.*, 2008] Yair Weiss, Antonio B. Torralba, and Robert Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems 21*, pages 1753–1760. MIT Press, 2008.

[Yu *et al.*, 2009] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2223–2231. 2009.

[Zhen and Yeung, 2012a] Yi Zhen and Dit-Yan Yeung. Co-regularized hashing for multimodal data. In *Advances in Neural Information Processing Systems 25*, pages 1385–1393. 2012.

[Zhen and Yeung, 2012b] Yi Zhen and Dit-Yan Yeung. A probabilistic model for multimodal hash function learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, 2012.