Probabilistic Semi-Supervised Multi-Modal Hashing

Behnam Gholami
bb510@cs.rutgers.edu

Computer Science Department
Rutgers, The state university of New Jersey,
New Brunswick, NJ, USA

Abolfazl Hajisami
hajisamik@cac.rutgers.edu

Department of Electrical and Computer Engineering
Rutgers, The state university of New Jersey,
New Brunswick, NJ, USA

Abstract

Learning hash functions for high dimensional multi-modal data is of great interest for many real-world retrieval applications in which data comes from diverse heterogeneous sources. In this paper, we propose a novel probabilistic semi-supervised multi-modal retrieval model, by which we can learn both the binary codes and their dimension from the available training data. We also develop a new Variational Bayes (VB) algorithm for learning the parameters of the proposed model. The experiments on two real-world data sets show the superiority of the proposed method over other state-of-the-art algorithms for learning binary codes.

1 Introduction

In recent years, multi-modal data has grown explosively due to the prevalence of social media (e.g., Youtube, Facebook, Flicker, etc). In such data, information comes from various sources such as images with textual descriptors, videos associated with audio signals, etc. Hence, each modality corresponding to a distinct input source can provide different types of information [1, 2, 15, 16, 26, 27].

Content-based multi-modal retrieval systems are very important to many applications of practical interest, such as obtaining relevant reviews and trailers of a movie using its poster or retrieving a set of images that best visually illustrate a given text. Such systems return the nearest neighbors of a given object query in the database based on the similarity between the query and of objects in that database.

A naive solution for searching the nearest neighbors requires the scan of all objects in the database that is obviously not scalable for large datasets with high dimensional feature vectors. On the other hand, the performance of approximate nearest neighbor searching algorithms such as tree-based methods is desirable for low-dimensional data, and their performance is unsatisfactory in the presence of high-dimensional feature vectors and does not guarantee faster search compared to linear scan [21].

Pages 28.1-28.12
DOI: https://dx.doi.org/10.5244/C.30.28

Among the various methods that have been proposed for nearest neighbor search [21], hashing-based methods have gained popularity in recent years [3, 5, 8, 10, 11, 14, 24, 35, 38, 39, 40]. The main advantage of hashing-based methods is that they encode data into binary features that lead to computational efficiency and low storage requirements [29, 33].

A broad range of hashing techniques can be grouped into unsupervised, semi-supervised, and supervised categories. Unsupervised hashing schemes aim at projecting nearby points in the original space to similar binary codes in the Hamming space (the space of binary codes) [6, 7, 22, 35].

On the other hand, supervised hashing methods map the original features to compact binary codes such that label based similarities are preserved in the Hamming space [10, 13, 14, 23, 29].

In semi-supervised hashing algorithms, the provided side information is available in binary mode (similar/dissimilar pairs). These algorithms try to find a projection of the data into the space of binary codes such that the Hamming distance between the codes reflects the similarity/dissimilarity relations between the similar/dissimilar pairs in the training set [14, 17].

Despite the increasing amount of multi-modal data, most existing hashing methods can only deal with unimodal data and consequently, they are not able to take advantage of the information of the various modalities for improving search accuracy and doing cross-view retrieval, e.g., text-to-image queries. Recently, some papers have been proposed to address this issue [12, 18, 25, 32, 36, 37]. Kumar and Udupa [12] extended spectral hashing [34] for cross-view similarity search. Nitish and Salakhutdinov proposed a deep Boltzmann machine for multi-modal feature learning and retrieval [25]. Rastegari *et al.* [20] utilized multiple SVMs for minimizing the Hamming distance between binary codes obtained from two different views. Very recently, Ozdemir and Davis proposed a probabilistic framework called IIBP for multi-modal retrieval using the Indian Buffet Process (IBP) model [18]. Using the IBP as a Bayesian nonparametric model, the IIBP is capable of learning both the binary codes and their dimension from the data.

Although the IIBP model is more accurate than other state of the art multi-modal retrieval methods, an important shortcoming of this model is that it cannot incorporate the information of the similarity/dissimilarity constraints into the posterior distribution of the parameters due to the use of Markov Chain Monte Carlo (MCMC) [18] algorithm.

To address this issue, in this paper, we propose a non-parametric Bayesian framework for multi-modal hash learning that takes into account the distance supervision (similarity/dissimilarity constraints). Our model embeds data of arbitrary modalities into a single latent binary feature with the ability to learn the dimensionality of the binary feature using the data itself. Given supervisory information (labeled similar and dissimilar pairs), we propose a novel discriminative term and develop a new Variational Bayes (VB) [31] algorithm which incorporates that term into the proposed Bayesian framework.

The rest of this paper is organized as follows. In Section 2, we present the proposed multi-modal hash learning framework. In Section 3, we introduce a novel VB algorithm to compute the posterior distribution of the parameters and the hidden variables. Experimental results are presented in Section 4. Finally, we conclude our work in Section 5.

2 Proposed Method

To facilitate the discussion, we assume we have two-modal data (e.g., images with both visual and textual descriptors), but our method can be easily extended to multi-modal data.

Let $\boldsymbol{T} = [\boldsymbol{X}, \boldsymbol{Y}]$ be the observed bi-modal data matrix where $\boldsymbol{X} = [x_1, x_2, ..., x_d]_{M \times d}$ and $\boldsymbol{Y} = [y_1, y_2, ..., y_d]_{N \times d}$ denote the first modal and the second modal data matrix respectively, and $\boldsymbol{Z} = [z_1, z_2, ..., z_d]_{K \times d}$ denotes the latent binary code matrix. We are also given two sets of pairwise constraints which are defined as

$$\mathcal{S} = \{(i,j) \mid (x_i, y_i) \text{ and } (x_j, y_j) \text{ are similar}\},$$
$$\mathcal{D} = \{(i,j) \mid (x_i, y_i) \text{ and } (x_j, y_j) \text{ are dissimilar}\},$$

where $\mathcal{S}(\mathcal{D})$ denotes the set of similar (dissimilar) pairwise constraints.

To be fully Bayesian, we must define appropriate prior and likelihood distributions for all observed $(\boldsymbol{X}, \boldsymbol{Y})$ and latent $(\boldsymbol{Z})$ variables. We assume each data point is independent of other data points given the latent binary codes and it is generated by a linear Gaussian model. More precisely, we have:

$$P(\boldsymbol{X}, \boldsymbol{Y} | \boldsymbol{Z}) = \prod_{i=1}^{d} \mathcal{N}(x_i; \boldsymbol{W_x} z_i, \gamma_x^{-1} \boldsymbol{I}) \mathcal{N}(y_i; \boldsymbol{W_y} z_i, \gamma_y^{-1} \boldsymbol{I}) \tag{1}$$

where $\boldsymbol{I}$ denotes the *Identity* matrix (the *identity* matrix is always assumed to be the appropriate size), $\boldsymbol{W_x} \in \mathbb{R}^{M \times K}$ and $\boldsymbol{W_y} \in \mathbb{R}^{N \times K}$ are the latent feature matrices, and γ_x and γ_y are the noise's precision values for each modality. In the above model, the elements of each binary code z_i can be considered as indicators of the possession of a corresponding column (feature) of $\boldsymbol{W_x}(\boldsymbol{W_y})$ for $x_i(y_i)$.

For learning both the latent binary codes and their dimension from data, we put a non-parametric prior distribution on the binary matrix $\boldsymbol{Z}$ by introducing auxiliary variables $\boldsymbol{\Pi} = \{\pi_k\}_{k=1}^{K}$ drawn from Beta distribution as

$$\pi_k \sim Beta(a/K, b(K-1)/K) \tag{2}$$

where a, b are the hyper-parameters and the integer K defines the largest possible dimension for the binary latent features (by letting $K \to \infty$, the dimensionality of the binary codes can be learned from the training data [28]). Then, we model the binary matrix $\boldsymbol{Z}$ as d draws from a Bernoulli process parameterized by $\boldsymbol{\Pi}$ that is generated as

$$z_i \sim \prod_{k=1}^{K} Ber(z_{ki}; \pi_k), \quad i = 1, ..., d \tag{3}$$

where z_{ki} denotes the k-th element of the binary vector z_i and Ber denotes the Bernoulli distribution (we obtain the IBP model by integrating out $\boldsymbol{\Pi}$ and letting $K \to \infty$). We also put conjugate prior distributions on the free parameters $\boldsymbol{W}_x, \boldsymbol{W}_y, \gamma_x$ and γ_y as

$$P(\boldsymbol{W}_x; \Sigma_x) \sim \prod_{k=1}^{K} \mathcal{N}(w_x^k; 0, \Sigma_x), \;\; P(\boldsymbol{W}_y; \Sigma_y) \sim \prod_{K=1}^{K} \mathcal{N}(w_y^k; 0, \Sigma_y), \tag{4}$$

$$P(\gamma_x; a_x, b_x) \sim Ga(a_x, b_x), \;\; P(\gamma_y; a_y, b_y) \sim Ga(a_y, b_y), \tag{5}$$

where Ga denotes the Gamma distribution, w_x^k and w_y^k denote the k-th column of the matrices $\boldsymbol{W_x}$ and $\boldsymbol{W_y}$ respectively and $a_x, b_x, a_y, b_y, \Sigma_x, \Sigma_y$ are the hyper-parameters of the proposed model. We show a graphical representation of the proposed model in Figure 1.

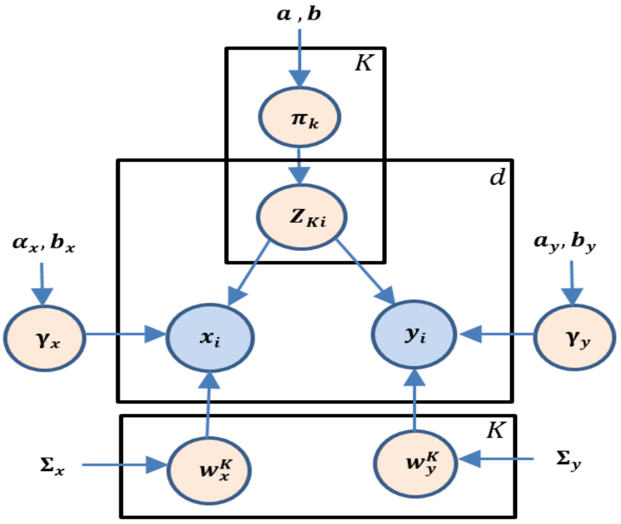

Figure 1: The graphical representation of the proposed Bayesian model (blue circles denote observations).

3 Posterior Inference

Since computing the exact posterior distribution of the parameters is intractable, in this section, we approximate that posterior distribution by developing a novel VB algorithm which incorporates the information of similarity/dissimilarity constraints into posterior distribution inference.

In our VB framework, we truncate the length of the binary codes (K) and we set it to a finite but large number. If K is large enough, the analyzed multi-modal data using this number of bits, will reveal less than K bits (see section 4).

Let $\Xi = [\boldsymbol{W_x}, \boldsymbol{W_y}, \boldsymbol{\Pi}, \gamma_x, \gamma_y]$ and $\boldsymbol{\Phi} = [a, b, a_x, b_x, a_y, b_y, \Sigma_x, \Sigma_y]$ denote the set of parameters and the set of hyper-parameters respectively, the joint probability of data and the unknown variables can be represented as

$$\begin{aligned} P(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \Xi \mid \Phi) = & \prod_{k=1}^{K} \prod_{i=1}^{d} P(z_{ki} \mid \pi_k) \prod_{k=1}^{K} p(w_x^k; \Sigma_x) p(w_y^k; \Sigma_y) P(\pi_k; a, b) \times \\ & \prod_{i=1}^{d} P(x_i \mid z_i, \boldsymbol{W_x}, \gamma_x) P(y_i \mid z_i, \boldsymbol{W_y}, \gamma_y) P(\gamma_x; a_x, b_x) P(\gamma_y; a_y, b_y) \end{aligned} \tag{6}$$

We use a fully factorized variational distribution for the latent variables as

$$q(\boldsymbol{\Pi}, \boldsymbol{Z}, \Xi) = \prod_{k=1}^{K} q_{\pi_k}(\pi_k) \prod_{i=1}^{d} \prod_{k=1}^{K} q_{z_{ki}}(z_{ki}) \prod_{k=1}^{K} q_{w_x^k}(w_x^k) \prod_{k=1}^{K} q_{w_y^k}(w_y^k) q_{\gamma_x}(\gamma_x) q_{\gamma_y}(\gamma_y) \tag{7}$$

Because of conjugacy between the prior and the likelihood distributions, we can easily specify the form of the approximate posterior distributions which goes as follows:

$$\begin{aligned} & q_{\pi_k}(\pi_k) = Beta(\pi_k; a_k, b_k),\ q_{z_{ki}}(z_{ki}) = Ber(z_{ki}; p_{ki}),\ q_{w_x^k}(w_x^k) = \mathcal{N}(w_x^k; \mu_x^k, \Sigma_x^k) \\ & q_{w_y^k}(w_y^k) = \mathcal{N}(w_y^k; \mu_y^k, \Sigma_y^k),\ q_{\gamma_x}(\gamma_x) = Gamma(a'_x, b'_x)\ q_{\gamma_y}(\gamma_y) = Gamma(a'_y, b'_y), \end{aligned} \tag{8}$$

where $\boldsymbol{\Theta} = \{p_{ki}, a_k, b_k, \mu_x^k, \Sigma_x^k, \mu_y^k, \Sigma_y^k, a'_x, b'_x, a'_y, b'_y\}$ denotes the set of posterior parameters which are determined by VB algorithm.

In order to incorporate the information of the similarity/dissimilarity constraints into the VB algorithm, we first define a regularizer for the binary code z_i as

$$\alpha(z_i) = \frac{1}{|\mathcal{D}_i|} \sum_{j:(i,j)\in\mathcal{D}} d(z_i, z_j) - \frac{1}{|\mathcal{S}_i|} \sum_{j:(i,j)\in\mathcal{S}} d(z_i, z_j) \tag{9}$$

where $d(z_i, z_j)$ denote the divergence (distance) function between two binary codes z_i and z_j, and $|\mathcal{S}_i|(|\mathcal{D}_i|)$ is defined as the number of data points which are similar (dissimilar) to the i-th data point. Intuitively, for each binary code z, $\alpha(z)$ should be large such that it best agrees with those constraints.

In a Bayesian framework, the parameters are random variables, and the notion of divergence between random variables can be replaced with the divergence between their corresponding posterior distributions. In our case, we use Kullback-Leibler (KL) divergence [9] to measure the distance between the posterior distributions of two binary codes. Hence, $\alpha(z_i)$ is written as

$$\alpha(z_i) = \frac{1}{|\mathcal{D}_i|} \sum_{j:(i,j)\in\mathcal{D}} KL(q_{z_i}(z_i)||q_{z_j}(z_j)) - \frac{1}{|\mathcal{S}_i|} \sum_{j:(i,j)\in\mathcal{S}} KL(q_{z_i}(z_i)||q_{z_j}(z_j)) \tag{10}$$

where $KL(p||q)$ denotes the KL divergence between two distributions p and q.

By defining the regularizer $\Omega(\boldsymbol{Z}) = \sum_{i=1}^{d} \alpha(z_i)$ for the binary code matrix $\boldsymbol{Z}$ using the set of similar/dissimilar pairs, the proposed regularized VB algorithm can be represented as the following optimization problem.

$$\hat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\mathbf{argmax}}\, \mathbb{E}_{q(\boldsymbol{\Pi},\boldsymbol{Z},\Xi)}[\log P(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \Xi \mid \Phi)] + H[q(\boldsymbol{\Pi}, \boldsymbol{Z}, \Xi)] + \lambda\Omega(\boldsymbol{Z}) \tag{11}$$

where $\mathbb{E}_q[.]$ denotes the Expectation operator respect to the distribution q, and $H[.]$ and λ denote the Entropy operator and the regularization parameter respectively. The VB algorithm simply solves the above optimization problem using the Coordinate Descent method. Since all distributions belong to the exponential family distributions, it can be shown that optimizing (11) with respect to the posterior distribution of each parameter corresponds to [31],

$$\log q(\Xi_i) = \mathbb{E}_{\Xi_{-i}}[\log P(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \Xi \mid \Phi)] + c, \tag{12}$$

where Ξ_i denotes the i-th posterior parameter, c is the summation of all terms which are independent of Ξ_i, and the expectation is taken over all the parameters except Ξ_i.

The update equation for each posterior parameter is as follows (we omit the details due to the lack of space). It is worth noting that if either the textual or the visual component is missing, we can simply integrate out the missing one by omitting the corresponding data term from the equations.

Update for $\boldsymbol{\Pi}$:

It is easy to show that a_k and b_k can be updated as

$$a_k = a/K + \sum_{i=1}^{d} p_{ki}, \quad b_k = b(K-1)/K + d - \sum_{i=1}^{d} p_{ki} \tag{13}$$

Update for $\boldsymbol{Z}$:

Because of the existence of the regularizer for the posterior parameters of the binary matrix $\boldsymbol{Z}$, we cannot use Eq. 12 to derive the update equation for the $q(\boldsymbol{Z})$. Instead, we reformulate the objective function of Eq. 11 as a function of the posterior parameters of $\boldsymbol{Z}$ and directly solve the induced optimization problem. It is worth to note that in expanding the regularizer $\alpha(z_i)$, we consider the second argument of $KL(.||.)$ to be a fixed deterministic value. It can be shown that we can update p_{ki} as $p_{ki} = \nu_{ki}/(\nu_{ki} + \nu'_{ki})$ where

$$\begin{aligned}\nu_{ki} =& \exp\Big\{\frac{a'_x}{2b'_x}\big(2x_i^\top \mu_x^k - ((\mu_k^x)^\top(\mu_k^x) + tr(\Sigma_x^k))\big)\Big\}\exp\Big\{\frac{a'_y}{2b'_y}\big(2y_i^\top \mu_y^k - ((\mu_k^y)^\top(\mu_k^y) + tr(\Sigma_y^k))\big)\Big\}\\ &\times \exp\{\langle\log \pi_k\rangle\} \times \exp\{\frac{\lambda}{|\mathcal{D}_i|}\sum_{j:(i,j)\in\mathcal{D}} \log p_{kj} - \frac{\lambda}{|\mathcal{S}_i|}\sum_{j:(i,j)\in\mathcal{S}} \log p_{kj}\} \qquad (14)\end{aligned}$$

and

$$\nu'_{ki} = \exp\{\langle\log(1-\pi_k)\rangle\}\exp\{\frac{\lambda}{|\mathcal{D}_i|}\sum_{j:(i,j)\in\mathcal{D}} \log(1-p_{kj}) - \frac{\lambda}{|\mathcal{S}_i|}\sum_{j:(i,j)\in\mathcal{S}} \log(1-p_{kj})\} \qquad (15)$$

where $tr(.)$ denotes the trace operator.

Update for $\boldsymbol{W_x}$ and $\boldsymbol{W_y}$:

In the following, we define $x_i^{-k} \equiv x_i - \langle\boldsymbol{W_x}\rangle_{-k}\langle z_i\rangle^{-k}$, and $y_i^{-k} \equiv y_i - \langle\boldsymbol{W_y}\rangle_{-k}\langle z_i\rangle^{-k}$, where $\langle\boldsymbol{W_x}\rangle_{-k}, \langle\boldsymbol{W_y}\rangle_{-k}$, and $\langle z_i\rangle^{-k}$ are the matrices/vector with the k-th column/element removed. μ_k^x and Σ_k^x are updated as

$$\Sigma_k^x = \left(\frac{a'_x}{b'_x}\sum_{i=1}^{d} p_{ki}\boldsymbol{I} + \Sigma^{-1}\right)^{-1}, \quad \mu_k^x = \Sigma_k^x\left(\frac{a'_x}{b'_x}\sum_{i=1}^{d} p_{ki}\langle x_i^{-k}\rangle\right) \qquad (16)$$

Similarly, μ_k^y and Σ_k^y are updated as

$$\Sigma_k^y = \left(\frac{a'_y}{b'_y}\sum_{i=1}^{d} p_{ki}\boldsymbol{I} + \Sigma^{-1}\right)^{-1}, \quad \mu_k^y = \Sigma_k^y\left(\frac{a'_y}{b'_y}\sum_{i=1}^{d} p_{ki}\langle y_i^{-k}\rangle\right) \qquad (17)$$

Update for γ_x, and γ_y:

It is easy to show that a'_x, b'_x can be updated as

$$a'_x = a_x + dM/2, \;\; b'_x = b_x + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{d}\beta_{ij}, \qquad (18)$$

where

$$\beta_{ij} = x_{ij}^2 - 2x_{ij}\sum_{k=1}^{K}(\mu_x^k)_i p_{kj} + \big(\sum_{k=1}^{K}(\mu_x^k)_i p_{kj}\big)^2 + \sum_{k=1}^{K}(\Sigma_x^k)_{ii}(p_{kj})(1-p_{kj}) \qquad (19)$$

where $(x)_i$ denotes the i-th element of the vector x and $(X)_{ii}$ denotes the i-th element on the main diagonal of the matrix X. Similarly, a'_y, b'_y is also updated in the same fashion.

Algorithm 1 Obtaining binary code for new observation

```
Require: $x^*, y^*, q(\boldsymbol{W_x}), q(\boldsymbol{W_y}), q(\gamma_x), q(\gamma_y), q(\Pi)$,
set $z = 0$ and index set $\mathcal{I} = \emptyset$.
for $k = 1, 2, \dots, K$ do
    set $\eta_k^+ = -\frac{a_x}{2b_x}\|x^* - \mu_x^k\|_2^2 - \frac{a_y}{2b_y}\|y^* - \mu_y^k\|_2^2 + \log\Gamma(a_k + 1) + \log\Gamma(b_k)$
    set $\eta_k^- = -\frac{a_x}{2b_x}\|x^*\|_2^2 - \frac{a_y}{2b_y}\|y^*\|_2^2 + \log\Gamma(a_k) + \log\Gamma(b_k + 1)$
end for
while $\max_k \eta_k^+ - \eta_k^- > 0$ do
    set $k' = \arg\max_k \eta_k^+ - \eta_k^-$, $\mathcal{I} \leftarrow \mathcal{I} \cup \{k'\}$, $z_{k'} = 1$, $\eta_k^+ = -\infty$
    for all $k \notin \mathcal{I}$ do
        set $\eta_k^+ = -\frac{a_x}{2b_x}\|x^* - \boldsymbol{\mu}_{\mathcal{I}}^{\boldsymbol{x}} - \mu_x^k\|_2^2 - \frac{a_y}{2b_y}\|y^* - \boldsymbol{\mu}_{\mathcal{I}}^{\boldsymbol{y}} - \mu_y^k\|_2^2 + \log\Gamma(a_k + 1) + \log\Gamma(b_k)$
        set $\eta_k^- = -\frac{a_x}{2b_x}\|x^* - \boldsymbol{\mu}_{\mathcal{I}}^{\boldsymbol{x}}\|_2^2 - \frac{a_y}{2b_y}\|y^* - \boldsymbol{\mu}_{\mathcal{I}}^{\boldsymbol{y}}\|_2^2 + \log\Gamma(a_k) + \log\Gamma(b_k + 1)$
    end for
end while
return $z$
```

Prediction for New Observations

Given $q(\boldsymbol{\Pi}, \Xi)$, the binary code z^* can be inferred for a new observation, (x^*, y^*) , using a MAP inference algorithm. Hence we maximize $\log P(z^*|x^*, y^*)$ by marginalizing out the variables $\boldsymbol{\Pi}, \boldsymbol{W_x}, \boldsymbol{W_y}, \gamma_x$, and γ_y:

$$z^* = \underset{z}{\textbf{argmax}} \log \int P(z|x^*, y^*, \boldsymbol{\Pi}, \boldsymbol{W_x}, \boldsymbol{W_y}, \gamma_x, \gamma_y)\, d\boldsymbol{\Pi}\, d\boldsymbol{W_x}\, d\boldsymbol{W_y}\, d\gamma_x\, d\gamma_y \tag{20}$$

Since the above integration cannot be computed in closed form, we approximate it by replacing $\boldsymbol{W_x}, \boldsymbol{W_y}, \gamma_x, \gamma_y$ with their posterior mean and integrating out $\boldsymbol{\Pi}$. Hence, the above optimization problem can be casted as:

$$\begin{aligned} z^* = \underset{z}{\textbf{argmax}} \; & -\frac{a'_x}{2b'_x}\|x^* - \boldsymbol{\mu}^{\boldsymbol{x}} z\|_2^2 - \frac{a'_y}{2b'_y}\|y^* - \boldsymbol{\mu}^{\boldsymbol{y}} z\|_2^2 + \sum_{k=1}^{K} \big(\log\Gamma(a_k + z_k) + \log\Gamma(b_k + 1 - z_k)\big), \\ & s.t. \;\; z \in \{0, 1\}^K \end{aligned} \tag{21}$$

where $\boldsymbol{\mu}^{\boldsymbol{x}} = [\mu_x^1, \dots, \mu_x^K]$, and $\boldsymbol{\mu}^{\boldsymbol{y}} = [\mu_y^1, \dots, \mu_y^K]$, and $\Gamma(.)$ denotes the *Gamma* function. Intuitively, the first two terms in Eq. 21 are data dependent terms, and the last term corresponds to the nonparametric penalty. More precisely, the value of the last term will be very small for low-probability $\boldsymbol{\Pi}$ elements, as learned through VB inference.

Since (21) is a combinatorial optimization problem, we use a greedy algorithm (Algorithm 1) similar to Orthogonal Maching Persuit (OMP) [30] to solve (21). In Algorithm 1, $\boldsymbol{\mu}_{\mathcal{I}}^*$ denotes the subvector of $\boldsymbol{\mu}^*$ formed by the dimensions indexed by $\mathcal{I}$. Intuitively, we initialize z with zero and sequentially set each entry of z to one, scoring each entry to determine which to set to one. As can be seen from Algorithm 1, the computational complexity of obtaining binary code for a query is $O(max(M + N) \times K^2)$.

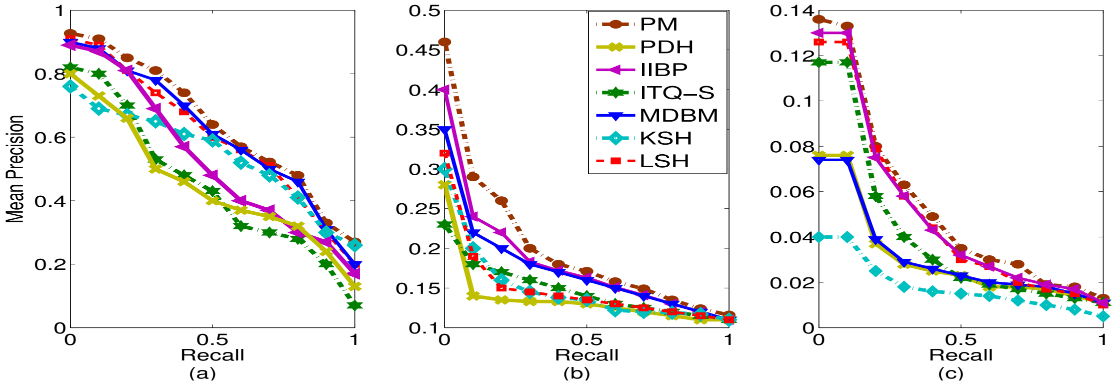

Figure 2: The result of category retrieval for text-to-image queries. (a) PASCAL-Sentence Dataset; (b) SUN Dataset (Euclidean ground truth computed from visual data); (c) SUN Dataset (Class label ground truth)

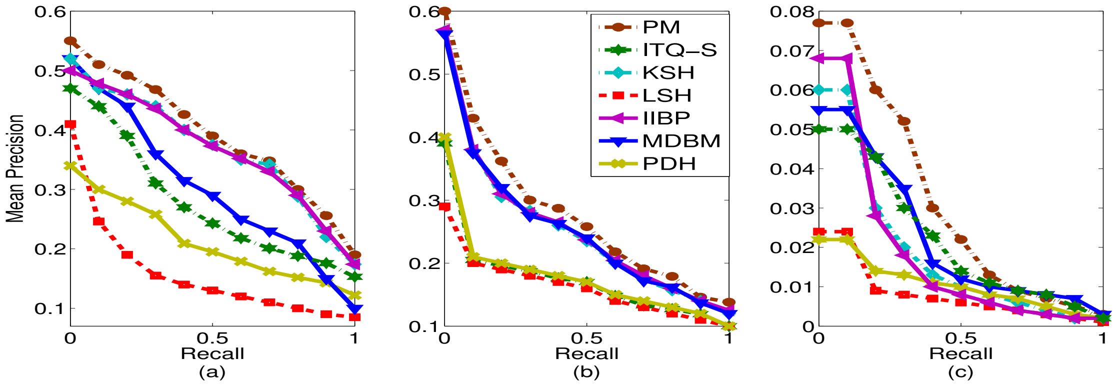

Figure 3: The result of category retrieval for image-to-image queries. (a) PASCAL-Sentence Dataset; (b) SUN Dataset (Euclidean ground truth computed from visual data); (c) SUN Dataset (Class label ground truth)

4 Experimental Results

We evaluate the proposed method on two benchmark bi-modal datasets: (1) The PASCAL-Sentence 2008 dataset [4] consists of 1000 images categorized into 20 classes. For this dataset, we used the precomputed visual and textual features provided by Farhadi *et al.* [4]. (2) The SUN-Attribute dataset [19] contains 102 attribute labels for each of the 14340 images from 717 categories. Following [18], for this dataset, we reduced the dimensionality of visual features from 19080 to 1000 by random feature selection. We also compute the attribute features by averaging the binary labels from multiple annotators.

We compare the performance of the proposed method (**PM**) against five state-of-the-art hashing methods, including four unsupervised methods locality sensitive hashing (**LSH**) [6], multi-modal deep Boltzmann machine (**MDBM**) [25], predictable dual view hashing (**PDH**) [20], and Integrated Indian Buffet Process (**IIBP**) [18] and two semi-supervised methods, Supervised Hashing with kernels (**KSH**) [14], and Iterative Quantization with supervised embedding (**ITQ-S**) [7]. Since **LSH**, **KSH**, and **ITQ-S** do not support cross-view queries, we applied them on single-view data. We initialized our model to use $K = 50$ number of bits for both datasets, though as the results show (Figure 1), only a small subset were ultimately

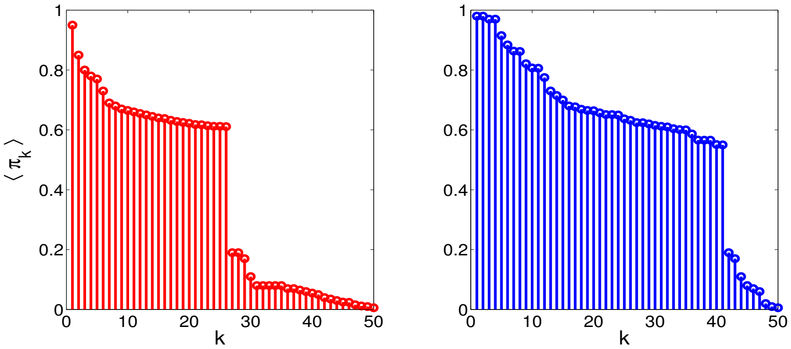

Figure 4: Left: Inferred $\langle \mathbf{\Pi} \rangle$ for the PASCAL-Sentence dataset, Right: Inferred $\langle \mathbf{\Pi} \rangle$ for the SUN-Attribute dataset.

used (other hashing methods were trained with the same code length K). We also randomly initialized the posterior parameters and run VB five times, selecting the run with the highest likelihood. The regularization parameter λ is tuned to an appropriate value on each dataset. All Gamma priors are set as $Ga(10^{-6}, 10^{-6})$ to make the prior distributions uninformative. The parameters a, b of the Beta distribution are set with $a = K$ and $b = K/2$ (many other settings of a and b yield similar results). We also set the hyper-parameters Σ_x/Σ_y to the empirical covariance of the visual/textual training data respectively. We use the Gaussian RBF kernel $k(x,y) = exp(-\|x-y\|_2^2/2\sigma^2)$ for **KSH** and the kernel parameter σ is tuned to an appropriate value on each dataset. We randomly select half of the data points for training and the other half as test set for both datasets. All images in the test segment were used as both image and text queries.

We need to generate side information in the form of pairwise training instances. We sample **similar** pairs by randomly selecting two instances from the same class and **dissimilar** pairs by choosing two instances from different classes. We randomly sample 20000 similar pairs and 20000 dissimilar pairs from the training set of each dataset.

For comparison purposes, we report the results of different methods via precision-recall curve as an accuracy measure. Since both datasets have multiple categories, we report mean precision and recall (by varying the number of top-retrieved samples, we can draw a precision-recall curve).

As can be seen from the Figures 2 and 3, Two major points can be inferred from the results. (i) Not surprisingly, the proposed method outperforms the other multi-modal hashing methods. The improvement in performance compared to **MDBM**, **PDH**, and **IIBP** is due to the fact that these methods do not have the similarity preserving property while **PM** can utilize the available similarity/dissimilarity side information. (ii) The proposed method has better performance than other semi-supervised hashing methods **KSH** and **ITQ-S** because they cannot exploit the correlation between two different visual and textual modalities of data.

In order to demonstrate the ability of the proposed method to learn the number of hash codes, we plot the sorted values of $\langle \mathbf{\Pi} \rangle$ for both datasets, inferred by the algorithm (Figure 4). As can be seen, the proposed VB algorithm inferred approximately 26 and 42 number of bits for PASCAL-Sentence and SUN-Attribute datasets respectively, fewer than the 50 initially provided.

5 Conclusion

We proposed a probabilistic semi-supervised binary hashing model for multi-modal data. The experimental results confirmed the improvements of our method over previous methods in the search accuracy of two multi-modal retrieval benchmark datasets. In our future work, we would like to develop a stochastic variational inference algorithm for binary feature learning on very large scale datasets.

References

[1] Mahdi Abavisani, Mohsen Joneidi, Shideh Rezaeifar, and Shahriar Baradaran Shokouhi. A robust sparse representation based face recognition system for smartphones. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6. IEEE, 2015.

[2] Behnam Babagholami-Mohamadabadi, Seyed Mahdi Roostaiyan, Ali Zarghami, and Mahdieh Soleymani Baghshah. Multi-modal distance metric learning: Abayesian nonparametric approach. In *European Conference on Computer Vision*, pages 63–77. Springer, 2014.

[3] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388. ACM, 2002.

[4] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010*, pages 15–29. Springer, 2010.

[5] Behnam Gholami and Abolfazl Hajisami. Kernel auto-encoder for semi-supervised hashing. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[6] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

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

[8] Qing-Yuan Jiang and Wu-Jun Li. Scalable graph hashing with feature transformation. IJCAI, 2015.

[9] James M Joyce. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, pages 720–722. Springer, 2011.

[10] Wang-Cheng Kang, Wu-Jun Li, and Zhi-Hua Zhou. Column sampling based discrete supervised hashing. In *AAAI*, 2016.

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

[12] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, volume 22, page 1360, 2011.

[13] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *ICML*, pages 1–8, 2011.

[14] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE, 2012.

[15] Shervin Minaee and Yao Wang. Screen content image segmentation using sparse decomposition and total variation minimization. *arXiv preprint arXiv:1602.02434*, 2016.

[16] Shervin Minaee and Yao Wang. Screen content image segmentation using robust regression and sparse decomposition. *arXiv preprint arXiv:1607.02547*, 2016.

[17] Yadong Mu, Jialie Shen, and Shuicheng Yan. Weakly-supervised hashing in kernel space. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3344–3351. IEEE, 2010.

[18] Bahadir Ozdemir and Larry S Davis. A probabilistic framework for multimodal retrieval using integrative indian buffet process. In *NIPS*, pages 2384–2392, 2014.

[19] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758. IEEE, 2012.

[20] Mohammad Rastegari, Jonghyun Choi, Shobeir Fakhraei, Daume Hal, and Larry Davis. Predictable dual-view hashing. In *ICML*, pages 1328–1336, 2013.

[21] Gregory Shakhnarovich, Piotr Indyk, and Trevor Darrell. *Nearest-neighbor methods in learning and vision: theory and practice*. 2006.

[22] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and Zhenmin Tang. Inductive hashing on manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1562–1569. IEEE, 2013.

[23] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. *CVPR*, 2015.

[24] SungRyull Sohn, Hyunwoo Kim, and Junm Kim. Supervised hashing via uncorrelated component analys. In *AAAI*, 2016.

[25] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2222–2230, 2012.

[26] Ali Taalimi, Shahab Ensafi, Hairong Qi, Shijian Lu, Ashraf A Kassim, and Chew Lim Tan. Multimodal dictionary learning and joint sparse representation for hep-2 cell classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 308–315. Springer, 2015.

[27] Ali Taalimi, Alireza Rahimpour, Cristian Capdevila, Zhifei Zhang, and Hairong Qi. Robust coupling in space of sparse codes for multi-view recognition. In *Image Processing (ICIP), 2016 IEEE International Conference on*, Sept 2016.

[28] Romain Thibaux and Michael I Jordan. Hierarchical beta processes and the indian buffet process. In *AISTATS*, pages 564–571, 2007.

[29] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *CVPR*, pages 1–8. IEEE, 2008.

[30] Joel Tropp, Anna C Gilbert, et al. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[31] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2): 1–305, 2008.

[32] Di Wang, Xinbo Gao, Xiumei Wang, and Lihuo He. Semantic topic multimodal hashing for cross-media retrieval. In *IJCAI*, pages 3890–3896, 2015.

[33] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

[34] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2009.

[35] Yair Weiss, Rob Fergus, and Antonio Torralba. Multidimensional spectral hashing. In *Computer Vision–ECCV 2012*, pages 340–353. Springer, 2012.

[36] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. Quantized correlation hashing for fast cross-modal search. In *IJCAI*, 2015.

[37] Liang Xie, Jialie Shen, and Lei Zhu. Online cross-modal hashing for web image retrieval. In *AAAI*, 2016.

[38] Bin Xu, Jiajun Bu, Yue Lin, Chun Chen, Xiaofei He, and Deng Cai. Harmonious hashing. pages 1820–1826. AAAI, 2013.

[39] Hanwang Zhang, Na Zhao, Xindi Shang, Huanbo Luan, and Tat-seng Chua. Discrete image hashing using large weakly annotated photo collection. In *AAAI*, 2016.

[40] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, 2016.