LOCALITY PRESERVING HASHING

Yi-Hsuan Tsai *Ming-Hsuan Yang*

Electrical Engineering and Computer Science
University of California, Merced
Merced, CA 95344, USA

ABSTRACT

The spectral hashing algorithm relaxes and solves an objective function for generating hash codes such that data similarity is preserved in the Hamming space. However, the assumption of uniform global data distribution limits its applicability. In the paper, we introduce locality preserving projection to determine the data distribution adaptively, and a spectral method is adopted to estimate the eigenfunctions of the underlying graph Laplacian. Furthermore, pairwise label similarity can be further incorporated in the weight matrix to bridge the semantic gap between data and hash codes. Experiments on three benchmark datasets show the proposed algorithm performs favorably against state-of-the-art hashing methods.

***Index Terms*—** Hashing, visual search, image retrieval

1. INTRODUCTION

Large-scale image and video retrieval has attracted much attention in recent years. Given a query, the simplest retrieval method is to determine the k nearest neighbors by sorting some similarity scores for the entire dataset. Obviously this approach is impractical when the number of data is large or a single similarity comparison is expensive. For effective and efficient large-scale image retrieval, numerous hashing algorithms have recently been introduced [1, 2, 3, 4, 5, 6, 7, 8]. The goal of hashing is to map high-dimensional data points to low-dimensional binary codes for efficient search of nearest neighbors in the Hamming space with linear time complexity.

One of the most effective approaches, the Locality-Sensitive Hashing (LSH) [9] algorithm uses random projections to construct hash codes. The underpinning of the LSH algorithm is that pairwise distance of data points can be theoretically guaranteed to be preserved in the Hamming space when the length of codes is sufficiently large. Kulis et al. propose a kernelized LSH (KLSH) method [10] for high-dimensional data when the feature embedding kernel is unknown. In addition, several techniques based on LSH have been developed for different distance metrics [11, 12]. On the other hand, several methods solve the hashing problem by optimizing different objective functions [13, 14]. One effective method proposed by Weiss et al. uses spectral relaxation [14] to compute hash codes. Compared to LSH, these approaches exploit data distribution to construct better hash codes where the pairwise distance is considered in the objective function and preserved. Although the above-mentioned unsupervised hashing approaches can be easily applied to different domains without the need to label data, some underlying important properties are not exploited and lead to the issues of semantic gap. Namely, when the data is mapped to hash codes, data points with the same semantic meanings may not have the same hash codes as the labels are not known or used.

Several supervised or semi-supervised hashing methods [12, 15, 16, 17] have been developed to exploit pairwise data labeled with similar or dissimilar attributes to leverage semantic similarity. Based on the LSH method, Kulis et al. use labeled samples to learn a Mahalanobis metric [12]. The semi-supervised hashing method [17] formulates an objective function which minimizes the empirical error of the labeled data while maximizing the entropy of the unsupervised term. Other methods such as binary reconstructive embeddings (BRE) [15] and minimal loss hashing [16] aim to learn hash codes by minimizing the reconstruction errors between the semantic space and Hamming space. While supervised methods have been demonstrated to achieve higher retrieval accuracy, they usually entail solving complicated optimization problems with high computational complexity.

In the paper, we focus on overcoming the limitations of the assumption that the data points are uniformly distributed. We propose a novel algorithm to exploit label information with an objective function that can be solved by spectral methods efficiently. Within the spectral hashing (SH) [14] framework, when approximating the eigenfunctions of the graph Laplacian of data points, the global structure of data points are modeled along the directions computed by principal component analysis (PCA) to satisfy the uniform distribution assumption. Since this assumption does not usually hold for real-world image data, we incorporate the Locality Preserving Projection (LPP) [18] method to exploit the local geometric structure between data points. Moreover, to bridge the semantic gap, we modify the weight matrix in LPP by adding a pairwise label similarity term to better enforce the semantic constraints of data points. We demonstrate the merits of the proposed algorithm with evaluations on three benchmark datasets with comparisons to state-of-the-art hashing methods.

2. PROPOSED ALGORITHM

To explore local relationship of data points, we use the LPP method [18] to project data along the directions that preserves local neighborhood information for estimating the eigenfunctions of the graph Laplacian. A spectral method with out-of-sample extension based on SH is then used to minimize the objective function for hashing. Furthermore, to preserve the semantic similarity between data points, we construct the pairwise label similarity matrix and incorporate it into the weight matrix of the LPP method.

2.1. Objective Function for Hashing

To compute effective binary hash codes, similar data points should be mapped to similar representations. Let $\{y_i\}_{i=1}^n$ denote the list of binary codes with length m for n data points, and $L = D - A$ be the Laplacian matrix, where A is the affinity matrix, and $D_{ii} = \sum_j A_{ij}$ is a $n \times n$ diagonal matrix. The affinity matrix is computed by $A_{ij} = \exp(-\|x_i - x_j\|^2/\epsilon^2)$, where $\{x_i\}_{i=1}^n$ is the original data list. The formulation to generate hash codes can be written in a matrix form:

$$\begin{aligned} &\min \operatorname{trace}(Y^\top L Y) \\ \text{subject to: } &Y(i,j) \in \{-1,1\} \\ &Y^\top 1 = 0 \\ &Y^\top Y = I, \end{aligned} \tag{1}$$

where Y is a $n \times m$ matrix whose i-th row is y_i. The constraint $Y^\top 1 = 0$ enforces each bit has a 50% chance to be one or zero, and the constraint $Y^\top Y = I$ enforces the bits are uncorrelated.

However, this formulation is equivalent to a graph partitioning problem which has been shown to be NP hard. To relax the problem in (1), the constraint $Y(i,j) \in \{-1,1\}$ can be removed, and the solution can be found by computing the m eigenvectors of L with minimal eigenvalues. Although the above problem can be solved efficiently after the relaxation, only the binary codes in the training set can be computed. Nevertheless, this out-of-sample problem can be solved by the approximation of eigenfunctions [14]. Assuming that the eigenfunctions of the one-dimensional Laplacian are uniformly distributed on $[a,b]$, the eigenfunctions $\phi_m(x)$ and eigenvalues λ_m are:

$$\phi_m(x) = \sin(\frac{\pi}{2} + \frac{k\pi}{b-a}x), \tag{2}$$

$$\lambda_m = 1 - e^{-\frac{\epsilon^2}{2}\|\frac{k\pi}{b-a}\|^2}. \tag{3}$$

2.2. Locality Preserving Hashing

In this work, we address two main issues of the current unsupervised hashing methods. First, when estimating the eigenfunctions of the graph Laplacian along PCA directions, only global structure of the data is captured rather than local neighborhood information. Second, while recent supervised hashing methods use pairwise label similarity to close the semantic gap, unsupervised methods do not exploit such information. Hence, we propose the Locality Preserving Hashing (LPH) method, which can be easily extended to the semi-supervised learning setting. The steps to compute LPP for the proposed LSH method are as follows:

1. **Construct the adjacency graph:** Let G denote the graph of the data, in which every node is connected to its k nearest neighbors to define edges.
2. **Compute the affinity matrix:** Let W be a matrix based on the graph G, $W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ if two nodes i and j are connected, and otherwise $W_{ij} = 0$.
3. **Compute the eigenmaps:** Compute the eigenvectors and eigenvalues for $XLX^\top$, where $L = D - W$ is the Laplacian matrix and $D_{ii} = \sum_j W_{ij}$.

In essence, the LPP method finds a linear mapping from nonlinear Laplacian eigenmaps, while preserving the local neighborhood information. From the directions constructed by LPP, we choose the m directions with the minimal eigenvalues to locally represent the data distribution for better estimation of eigenfunctions.

In this work, the weight matrix W is incorporated with the pairwise label similarity to better describe the affinity of data points:

$$W_{ij} = (1-\lambda)\exp(-\|x_i - x_j\|^2/2\sigma^2) + \lambda S_{ij}, \tag{4}$$

where S is the label similarity matrix and λ controls the importance of pairwise label similarity. We define S_{ij} is equal to 1 if x_i and x_j have the same label, and otherwise it is equal to 0.

The weight matrix in (4) does not only consider the data distribution but also the semantic similarity. It penalizes the cases where two data points with the same label happen to have large distance. Note that as we only use a subset of labeled data, our approach with the pairwise similarity can be considered as a semi-supervised method, but can be computed efficiently as unsupervised hashing methods. The main steps of our algorithm are summarized as follows:

1. **Compute LPP:** Use LPP to find directions that preserves local relation of data points.
2. **Compute eigenfunctions:** Along each LPP direction, approximate the m eigenfunctions of the graph Laplacian with the m smallest eigenvalues using (2).
3. **Compute hash codes:** Threshold the eigenfunctions at zero to obtain the final binary codes.

2.3. Discussion

To gain more insight on how LPH works with LPP, we discuss the relationship between PCA and LPP. Let $W_{ij} = 1/n^2$, we show that $XLX^\top$ in the third step of LPP is equal to the covariance matrix for computing PCA.

$$\begin{aligned}
XLX^\top &= X(\frac{1}{n}I - \frac{1}{n^2}ee^\top)X^\top \\
&= \frac{1}{n}XX^\top - \frac{1}{n^2}(Xe)(Xe)^\top \\
&= \frac{1}{n}\sum_i x_i x_i^\top - \frac{1}{n^2}(nm)(nm)^\top \\
&= \frac{1}{n}\sum_i x_i x_i^\top - 2mm^\top + mm^\top \\
&= E[(x-m)(x-m)^\top],
\end{aligned}$$

where $m = \frac{1}{n}\sum_i x_i$ is the data mean, e is a vector with all elements of 1. This shows that the weight matrix W plays a key role in LPP. When the graph G is fully connected with the proper W_{ij}, $XLX^\top$ is the covariance matrix of the data.

Another way to analyze LPP is to determine the value of k for nearest neighbors when constructing the graph G. If k is large, W_{ij} tends to preserve the global structure (as PCA), and if k is sufficiently small, the data is projected along the directions preserving locality which have the minimal local variance.

3. EXPERIMENTAL RESULTS

We conduct image retrieval experiments on the CIFAR-10, CIFAR-100 and PASCAL VOC 2010 datasets, and evaluate our proposed method against state-of-the-art methods. In addition, we compare the results with and without using the proposed pairwise similarity matrix (i.e., in semi-supervised or unsupervised settings). For quantitative evaluation, we adopt the commonly used Hamming ranking with 8 to 64 bit hash codes. Given a query image, all the data points are ranked according to their Hamming distance to the query. We report the mean precision and precision-recall curves of Hamming ranking with comparisons to SH [14], KLSH [10], and BRE [15] algorithms using the original implementations.

CIFAR-10. The CIFAR-10 dataset is a subset of 80 million tiny images [19] with labels. It consists of 60K 32×32 images of 10 object categories represented by 512-dimensional GIST features. The entire dataset is uniformly partitioned from 10 classes into a training set of 59K images, and a test set of 1K query images. We select 200 images from each class randomly for samples for methods involving kernels. The pairwise label similarity is also obtained from this 2K training subset. We use $k = 100$ for nearest neighbors for experiments with this dataset.

CIFAR-100. The CIFAR-100 dataset is also a labeled subset of 80 million tiny images, containing 100 classes with 600 images each. The entire dataset is uniformly partitioned from 100 classes into a training set of 59K images, and a test set of 1K query images. For the training subset, 20 images from each class are randomly selected with a total of 2K samples. The same parameters are used as for the CIFAR-10 dataset.

PASCAL VOC 2010. The PASCAL VOC 2010 [20] dataset is widely used for object recognition tasks. It contains 20 classes with about 10K images where each one has one or more labels from different object categories. Different from the other two datasets, it has more complicated variations of object appearance with view-point changes, occlusions, and cluttered background regions.

The entire dataset is uniformly partitioned from 20 classes into a training set of around 9k images, and the remaining ones are used for tests. For the training subset, 20% of images from the training set are randomly selected. To better account for large appearance change across images, we extract features using bag-of-words SIFT histograms with 1024 dimensions for each image, and use $k = 500$ for nearest neighbors to construct the weight matrix.

Results. As shown in Fig. 1, the proposed LPH algorithms, with and without using labeled data, perform favorably against other hashing algorithms in all three datasets. The LSH and KLSH methods perform worse than all the other methods, which can be attributed to the fact that LSH-based hashing methods usually require longer hash codes for accuracy due to the use of random projections. We use $\lambda = 0.9$ to control the weight for label similarity S in all the experiments and the proposed LPH method performs better when the weight matrix W is constructed using (4). This also demonstrates the label similarity facilitates closing the semantic gap of data points and hash codes as the semantic similarity is more reliable than data similarity.

For the CIFAR-10 dataset, the proposed LPH method performs well (with and without label similarity) against the SH, KLSH, and LSH methods. However, in the precision-recall curve (Fig. 1(b)), the LPH method without using labeled data performs similarly to the one using pairwise label similarity. Unlike other semi-supervised hashing methods which use labeled data in the objective function, our label information is only used for adjusting the data distribution and estimating the eigenfunctions. In some simpler datasets such as CIFAR-10, the LPH method does not gain additional information from labeled data. On the other hand, the proposed LPH algorithm using labeled data performs best in the challenging CIFAR-100 dataset since it has more object classes and the label similarity is likely to play a more important role. The merits of semi-supervised LPH approach can be clearly observed in the precision-recall curve of Fig. 1(d).

For the PASCAL VOC 2010 dataset, both proposed meth-

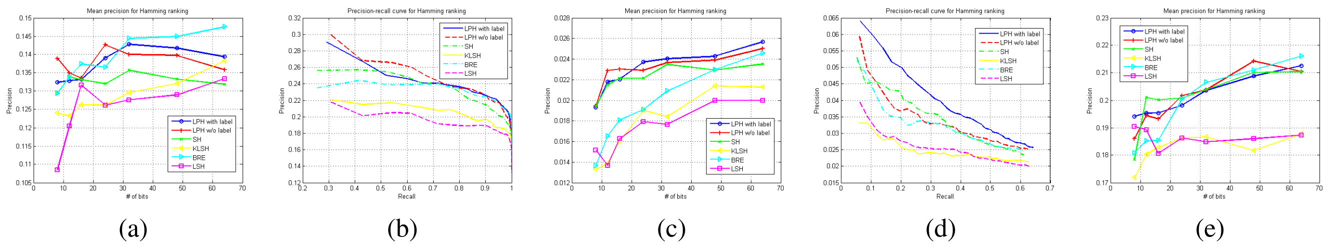

Fig. 1. (a) CIFAR-10: precision of the top 500 returned samples; (b) CIFAR-10: precision-recall curve with 64 bits; (c) CIFAR-100: precision of the top 50 returned samples; (d) CIFAR-100: precision-recall curve with 64 bits; (e) VOC 2010: precision of the top 50 returned samples.

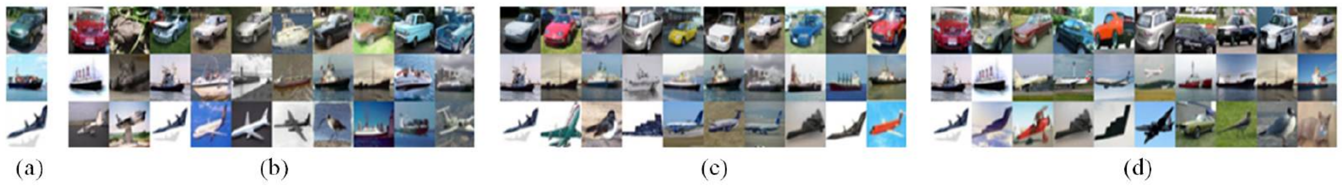

Fig. 2. Qualitative results on CIFAR-10 with top 10 returned images using 64 bits. (a) Query images; (b) LPH with labels; (c) LPH without labels; (d) SH.

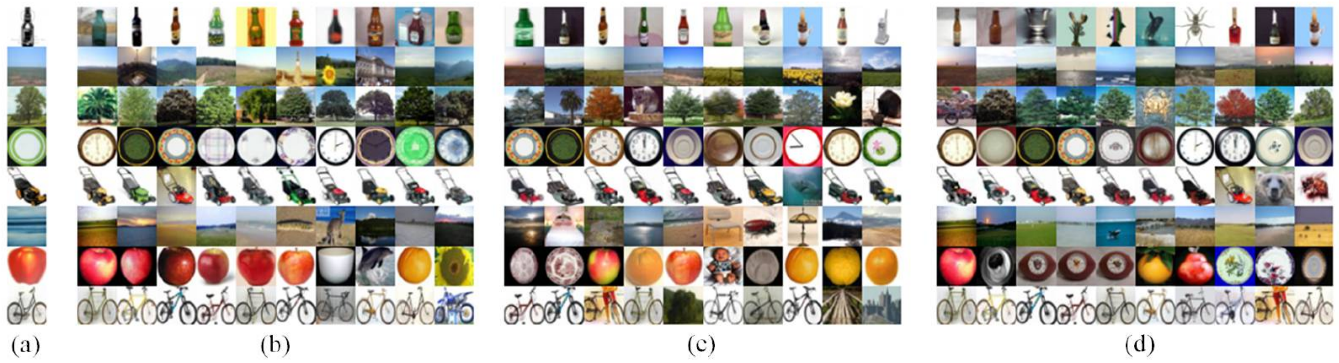

Fig. 3. Qualitative results on CIFAR-100 with top 10 returned images using 64 bits. (a) Query images; (b) LPH with labels; (c) LPH without labels; (d) SH.

ods perform well against the SH and BRE methods (Fig. 1(e)), but LPH tends to perform better than BRE when the hash code length is small. This results suggest that by using LPP, especially for images with large variations, the m directions (bits) with minimal eigenvalues can preserve most of the energy for data's local distribution. Note that since each image may have multiple labels, precision is computed based on whether the query and the returned images share at least one common label (same definition as the pairwise label similarity). Fig. 2 and Fig. 3 present some retrieval results using different hashing methods. The proposed LPH method with labels retrieve more consistent results than LPH without labels and the SH method. More results and large images are available at `https://sites.google.com/site/yihsuantsai/research/lph/`.

4. CONCLUDING REMARKS

In this paper, we propose a hashing algorithm which accounts for the local relationship between data points using LPP. The proposed method is extended in a semi-supervised fashion by incorporating the pairwise label similarity in the weight matrix. The proposed LPH method with labels can be easily trained via a spectral solver. Experimental results show that the proposed algorithms, with and without label information, perform favorably against state-of-the-art hashing methods on three benchmark datasets.

Our future work includes exploring other methods to approximate eigenfunctions that are not limited to the uniform data assumption. In addition, it is of great interest to exploit local structural information for supervised hashing methods.

5. REFERENCES

[1] RS Lin, DA Ross, and Jay Yagnik, “Spec hashing: Similarity preserving algorithm for entropy-based coding,” in *CVPR*, 2010.

[2] Wei Liu, J Wang, Y Mu, S Kumar, and SF Chang, “Compact hyperplane hashing with bilinear functions,” in *ICML*, 2012.

[3] Wei Liu, Jun Wang, Rongrong Ji, and Yu-gang Jiang Shih-fu Chang, “Supervised Hashing with Kernels,” in *CVPR*, 2012.

[4] Yue Lin, Rong Jin, Deng Cai, Shuicheng Yan, and Xuelong Li, “Compressed hashing,” in *CVPR*, 2013.

[5] Yang Liu, Fei Wu, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann, “Spline regression hashing for fast image search.,” *TIP*, vol. 21, no. 10, pp. 4480–91, 2012.

[6] Yadong Mu, John Wright, and SF Chang, “Accelerated Large Scale Optimization by Concomitant Hashing,” in *ECCV*, 2012.

[7] Mohammad Norouzi, David J Fleet, and Ruslan Salakhutdinov, “Hamming Distance Metric Learning,” in *NIPS*, 2012.

[8] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon, “Spherical hashing,” in *CVPR*, 2012.

[9] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, “Similarity search in high dimensions via hashing,” in *VLDB*, 1999.

[10] Brian Kulis and Kristen Grauman, “Kernelized locality-sensitive hashing,” *PAMI*, vol. 34, no. 6, pp. 1092–104, 2012.

[11] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *SOCG*, 2004.

[12] Brian Kulis, Prateek Jain, and Kristen Grauman, “Fast similarity search for learned metrics.,” *PAMI*, vol. 31, no. 12, pp. 2143–57, 2009.

[13] Ruslan Salakhutdinov and Geoffrey Hinton, “Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure,” in *AISTATS*, 2007.

[14] Yair Weiss, Antonio Torralba, and Rob Fergus, “Spectral hashing,” in *NIPS*, 2008.

[15] Brian Kulis and Trevor Darrell, “Learning to hash with binary reconstructive embeddings,” in *NIPS*, 2009.

[16] M Norouzi and DJ Fleet, “Minimal loss hashing for compact binary codes,” in *ICML*, 2011.

[17] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang, “Semi-Supervised Hashing for Large Scale Search.,” *PAMI*, vol. 6, no. 1, pp. 1–14, 2012.

[18] Xiaofei He and Partha Niyogi, “Locality preserving projections,” in *NIPS*, 2003.

[19] Antonio Torralba, Robert Fergus, and William T. Freeman, “80 million tiny images: A large data set for non-parametric object and scene recognition.,” *PAMI*, vol. 30, no. 11, pp. 1958–1970, 2008.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.