

Supervised Topology Preserving Hashing

Shu Zhang^{1,2} Man Zhang^{1,2} Qi Li^{1,2} Tieniu Tan^{1,2,3} Ran He^{1,2,3}

¹Center for Research on Intelligent Perception and Computing, CASIA

²National Laboratory of Pattern Recognition, CASIA

³Center for Excellence in Brain Science and Intelligence Technology, CAS

{shu.zhang, zhangman, qli, tnt, rhe}@nlpr.ia.ac.cn

Abstract

Learning based hashing is gaining traction in large-scale retrieval systems. It aims to learn compact binary codes that can preserve semantic similarity in the hamming space. This paper presents a supervised topology hashing (STPH) algorithm to learn compact binary codes that can exploit both the supervisory information as well as the local topology structure of datasets. To build a connection between the original space and the resultant hamming space, we minimize the quantization errors together with a classification error term and a topology preserving term. A nonlinear kernel feature space is further used to improve the generalization power. An alternating iterative algorithm is developed to minimize the complex objective function that contains both continuous and discrete variables. Experimental results on three benchmark datasets demonstrate the effectiveness of the proposed method on image retrieval tasks.

1. Introduction

Hashing techniques aim to map high-dimensional data to the hamming space while preserving some predefined similarity in the original space (e.g., Euclidean space or semantic space). Due to computational efficiency of the hamming distance and the low storage overhead to store binary codes, hashing has become one of the most popular approximate nearest neighbor (ANN) search techniques for many computer vision applications, including content-based image retrieval [10], image matching [17] and object recognition [18, 6].

Most of the existing hashing techniques can be generally categorized into two classes: data-independent[4, 9] and data-dependent[5, 8, 12, 15, 21]. For data-independent hashing, locality sensitive hashing (*LSH*) [4] and its variants [9] are representative methods, which employ randomly generated projections as their hashing functions. For the second category, various statistical learning tech-

niques are exploited to learn hashing functions with or without supervisory information. State-of-the-art unsupervised hashing methods in this category include spectral hashing (*SH*) [20], iterative quantization (*ITQ*) [5], and hashing with graphs (*AGH*) [13]. In contrast, various hashing techniques leverage the supervisory information for better semantic neighbor search, including *CCA-ITQ* [5], binary reconstructive embedding (*BRE*) [8], kernel supervised hashing (*KSH*) [12], etc. Recently, topology preserving hashing (*TPH*) [21] has been proposed to utilize the local topology information to improve the performance of hashing. In addition to preserving the neighborhood relationship[12, 13, 20], *TPH* preserves the topological structures via preserving the neighborhood ranking [21].

Inspired by topology preserving hashing [21], this paper presents a supervised topology preserving hashing (*STPH*) algorithm to learn compact binary codes for large scale visual search. Different from *TPH*, *STPH* aims to learn discriminative hashing function by fully exploiting supervisory information as well as preserving topological structures. One classification error term and one topology preserving term is thereby proposed to construct our cost function for hash function learning. In order to connect the original high-dimensional space and the resultant hamming space, an quantization error term is explicitly added to the overall objective function so as to minimize the quantization loss like they did in[5]. To further improve the generalization power, we seek linear hash functions in a nonlinear kernel feature space as in [12].

There are two major contributions of this work: 1) We develop a supervised hashing technique that exploit both supervisory label information and data topology structure to learn compact binary codes for large scale visual search. By preserving both the semantic and topology structure, we were able to learn binary codes that can provide a good result ranking. 2) An alternating iterative algorithm is derived to efficiently optimize the proposed objective function which involves both continuous and discrete variables. Experimental results on three benchmark datasets show that

the proposed method can significantly improve the discriminability of *TPH* and obtain better retrieval results than several state-of-the-art hashing methods.

2. The Proposed Approach

In this section, we present the formulation of our hashing method, *i.e.*, *supervised topology preserving hashing (STPH)*. In particular, we are given a set of N points $X = \{x_i\}, i = 1 \dots N, X \in \mathbb{R}^{d \times N}$ and their corresponding label stored in matrix $Y \in \mathbb{R}^{c \times N}$. c is the total number of classes and the position of 1 in each column $y_i = [0 \dots 1 \dots 0]^T$ denotes the correct class label. The goal of *STPH* is leverage both the label information and the topology structure so as to learn hash functions that can preserve both the semantic and topology structure of the original space. Unlike the previous mentioned semi-supervised extensions of *TPH* which only use label information to construct the Topo-weighting matrix [21], we formulate the following minimization problem to jointly learn a projection matrix P , the resultant binary codes B and a classifier W . The cost function consists of four parts.

$$\begin{aligned} \min_{P, W, B} J &= J_1 + J_2 + J_3 + J_4 \\ &= \|Y - W^T B\|_F^2 + \alpha \|B - P^T X\|_F^2 \\ &\quad + \beta \|W\|_F^2 + \gamma \text{tr}(P^T X L X^T P) \\ \text{s.t. } B &\in \{-1, 1\}^{k \times N} \end{aligned} \quad (1)$$

where $B \in \{-1, 1\}^{k \times N}$ is the binary codes with each column representing one sample. $P^T \in \mathbb{R}^{k \times d}$ is the projection matrix that transform the zero-centered features $X \in \mathbb{R}^{d \times N}$ in the original space to the k -bit binary embedding with $B = \text{sgn}(P^T X)$. $Y \in \{0, 1\}^{c \times N}$ is the label ground-truth. The matrix $L \in \mathbb{R}^{N \times N}$ is the Laplacian matrix [1].

The problem in (1) aims to minimize the empirical error (J_1) of classifying the binary codes B to their corresponding class label Y [7] while keeping the quantization error (J_2) introduced in the *sgn* step as small as possible. For ease of computation, the Frobenius norm instead of the more sophisticated maximum correntropy criterion [2] is employed to measure the quantization error. J_3 is a regularization term for large W . And J_4 penalizes P to preserve the topology structure of the original space during the hashing projection. Therefore, the overall cost function seeks projection matrix P that can preserve the topology structure of the original space, and make the learnt binary codes as discriminative as possible. It is worth noting that J_4 comes in the form of $\text{tr}(P^T X L X^T P)$ instead of $\text{tr}(B L B^T)$, where we follow a common practice [20] and relax B with its signed magnitude $P^T X$, since

$B = \text{sgn}(P^T X)$. Unlike the convention in previous literature, we directly optimize for B in our cost function and employ J_2 as a connection between the projection matrix P and the resultant binary codes B . From the next section, we can see that our formulation of *STPH* in (1) give rise to a very efficient and effective optimization process, validating its correctness.

Moreover, for better generalization performance, we use the kernel feature $\phi(x)$ generated with an RBF kernel mapping process: $\phi(x) = [\exp(\|x - x_1\|^2/\sigma), \dots, \exp(\|x - x_h\|^2/\sigma)]$, where $\{x_i\}_{i=1}^h$ are h chosen anchor points from the training samples and σ is the kernel width. Typically, the anchor points can either be chosen as the clustering centers with k-means [13] or like we do in this work, as randomly chosen samples. In the next section, we represent the kernel feature $\phi(x)$ of all data points as matrix X , for simplicity.

2.1. Optimization

To make the non-convex optimization problem in Equation (1) tractable, we employ an alternating optimization procedure, where we minimize the problem with respect to one variable while fixing others at each step and iterate over all the steps. We address the detailed optimization process in the following paragraph.

Firstly, the features X in the kernel space should be computed as mentioned in the previous section, and B is initialized with randomly generated binary codes. Then the algorithm iterates over the following steps to minimize the objective function in Equation (1).

W-Step By fixing all the other variables except for W , this problem degenerates to a least squares problem with a closed-form solution:

$$W = (BB^T + \beta I)^{-1} BY^T \quad (2)$$

P-Step While fixing all the variables except for P , the degenerated problem comes in the following form:

$$\begin{aligned} \arg \min_P \alpha \|B - P^T X\|_F^2 + \gamma \text{tr}(P^T X L X^T P) \\ \text{s.t. } B \in \{-1, 1\}^{k \times N} \end{aligned} \quad (3)$$

The problem in (3) is convex with respect to P . Therefore, by setting the derivative to zero, a global minimum of P in this sub-step can be computed as follows:

$$P = (2\alpha XX^T + \gamma X(L + L^T)X^T)^{-1} 2\alpha XB^T \quad (4)$$

B-Step In this step, we aim to optimize B with all other variables fixed. The optimization problem in this step takes the following form:

$$\begin{aligned} \arg \min_B \|Y - W^T B\|_F^2 + \alpha \|B - P^T X\|_F^2 \\ \text{s.t. } B \in \{-1, 1\}^{k \times N} \end{aligned} \quad (5)$$

Table 1. Results on the CIFAR-10 dataset with regards to different number of bits. The first two columns show the Hamming ranking results evaluated by mAP and precision@500. The right column shows the Hamming look-up results when the Hamming radius $r = 2$.

Method	mAP			precision@500			precision@r=2		
	12	24	36	12	24	36	12	24	36
CCA-ITQ	30.22	33.90	35.07	39.85	41.40	43.65	34.24	40.93	36.29
KSH	34.60	39.71	42.94	42.15	46.90	50.10	39.30	40.95	27.72
FastHash	35.66	41.99	43.71	31.70	40.43	41.67	35.95	25.34	8.66
SSH	17.04	19.11	19.97	16.42	22.62	26.60	14.94	20.87	23.75
BRE	14.02	14.67	14.55	16.70	22.60	23.30	11.23	15.93	18.55
PCA-ITQ	17.02	17.68	17.68	25.09	28.11	29.22	21.08	18.95	5.21
TPH	18.21	17.71	18.24	27.70	29.52	31.02	22.81	19.33	5.54
STPH	38.66	43.57	44.42	45.64	47.83	50.17	39.53	47.03	41.29

By expanding the Frobenius norm in (5) and removing the constants, we get the following optimization problem:

$$\begin{aligned} & \arg \min_B \|W^T B\|_F^2 - 2 \operatorname{tr}(B^T Q) \\ & \text{s.t. } B \in \{-1, 1\}^{k \times N} \end{aligned} \quad (6)$$

Where $Q = WY + \alpha P^T X$. Although the resultant problem is NP hard, we can iteratively learn B bit by bit due to the separable property of inner products [15]. Now, we introduce some notations for a clear description of the following optimization step. Let b^T be the l^{th} row of B , $l = 1, \dots, k$ and B' the matrix of B excluding b . Similarly, let q^T be the l^{th} row of Q , Q' the matrix of Q excluding q , w^T the l^{th} row of W and W' the matrix of W excluding w . Then we can expand problem (6) and derive an equivalent problem as follows:

$$\arg \min_b (w^T W'^T B' - q^T b) \quad \text{s.t. } b \in \{-1, 1\} \quad (7)$$

because:

$$\begin{aligned} \|W^T B\|_F^2 &= \operatorname{tr}(B^T W W^T B) \\ &= \text{const} + \|bw^T\|_F^2 + 2w^T W'^T B' b \\ &= \text{const} + 2w^T W'^T B' b \end{aligned} \quad (8)$$

and similarly $\operatorname{tr}(B^T Q) = \text{const} + q^T b$.

The optimal solution of problem (7) is:

$$b = \operatorname{sgn}(q - B'^T W' w) \quad (9)$$

In practice, the B-step should iterate over all the bits with (9) for about 2 to 5 times to yield a stable B .

3. Experiments

We present quantitative evaluations in terms of several retrieval metrics and compare STPH with unsupervised methods: PCA-ITQ [5], IMH [16] and supervised methods: FastHash [11], CCA-ITQ [5], BRE [8], SSH [19],

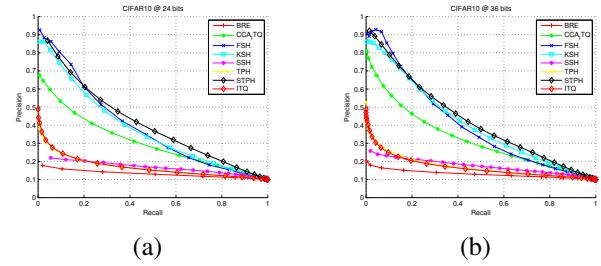


Figure 1. The precision recall curve on the CIFAR-10 dataset with regards to (a) 24 bits and (b) 36 bits.

KSH [12], semi-supervised extension of TPH [21]. To perform fair evaluation, we adopt two criteria commonly used in the literature, *i.e.*, Hamming ranking and hash lookup performance. Specifically, four evaluation metrics are used to measure the performance in total. In particular, for Hamming ranking based evaluation, we report the retrieval precision of the top 500 returned samples, the mean of average precision (mAP) and the precision-recall curve. And with respect to hash lookup performance, the precision of the returned samples falling within Hamming radius 2 is reported.

In our experiments, three image datasets are used, *i.e.*, CIFAR-10¹, MNIST² and NUS-WIDE³. The CIFAR-10 dataset consists of 60K commonly seen images which are manually categorized into 10 classes (6K samples per class). Each image is represented with a 512 dimension GIST [14] vector feature. The MNIST dataset consists of ten handwritten digits ranging from '0' to '9', each with 7K 28 × 28 grayscale images. For MNIST and CIFAR-10, following a common procedure, a query set consists of 1000 samples is sampled uniformly from the whole dataset, and 5000 labeled images are used as both the training and gallery set for evaluation efficiency. Neighborhood ground-truth are defined by the correct label information from the datasets. The NUS-WIDE dataset [3] is a set of Flickr consumer images collected by NUS lab that contains around

¹<http://www.cs.toronto.edu/~kriz/cifar.html>

²<http://yann.lecun.com/exdb/mnist/>

³<http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm>

Table 2. Results on the MNIST dataset with regards to different number of bits. The first two columns show the Hamming ranking results evaluated by mAP and precision@500. The right column shows the Hamming look-up results when the Hamming radius $r = 2$.

Method	mAP			precision@500			precision@r=2		
	12	24	36	12	24	36	12	24	36
CCA-ITQ	72.79	76.48	77.49	80.15	81.80	83.05	75.25	79.30	72.74
KSH	88.72	91.65	91.49	88.62	91.30	91.18	86.97	88.90	84.93
FastHash	86.24	88.71	90.21	85.57	88.87	89.53	85.24	83.99	73.50
SSH	21.46	45.45	34.32	22.82	48.76	51.76	17.69	39.58	43.47
BRE	10.11	41.16	36.78	13.70	77.40	78.80	9.84	72.64	48.64
PCA-ITQ	37.96	41.26	42.46	69.21	78.65	81.26	52.12	71.17	37.11
TPH	43.16	42.84	43.97	71.73	79.82	83.91	56.98	71.89	37.48
STPH	87.59	92.21	92.53	89.83	92.47	92.76	86.73	90.90	87.91

270,000 images associated with 81 ground truth concept tags, with each image assigned to multiple semantic labels. Since all the compared methods mainly focus on the traditional visual search problem, *i.e.*, searching images with mutually exclusive labels. Therefore, we select a subset that belong to the 21 largest classes with each image exclusively belonging to one of the 21 classes, which results in a subset with 72,219 images. It is noted that different classes have different number of images. The images in the dataset are represented with 500-dim SIFT feature. For each class, 1/10 of the images are sampled as the query set and the remaining images are used as the training and gallery set. Since this dataset is relatively larger, the training time of *BRE* and *FSH* will take many hours. For evaluation efficiency, we only compare *STPH* with *CCA-ITQ* and some other very efficient unsupervised methods. For parameters in *STPH*, we empirically set α around 1e-3, γ to 1 and β to 1e-1. The number of anchor points for all the kernel based methods is fixed to 1000 for fair comparison.

3.1. Results

We report detailed quantitative evaluation results with 12, 24 and 36 bits in Table 1, 2, 3 and precision recall curve in Figure 1, 2, 3. It is obvious that our *STPH* achieves the best results in most cases compared to all the competing methods. Since our approach takes the full label information into consideration and employ the nonlinear kernel feature, it is not surprising to see that our approach consistently outperforms the semi-supervised *TPH*. And because the incorporation of the topology-preserving term, our approach also achieves better performance than supervised methods like *KSH* and *FastHash*.

Since they take topology preserving into consideration in their framework, *TPH* always achieves slightly better results than the benchmark method *ITQ* (or *PCA-ITQ*). Moreover, compared to the *TPH*, the proposed *STPH* achieves far better results as we explicitly exploit supervisory label information in our formulation as well as preserve the topology structure. Among all the methods that achieves good results, *KSH* also learn their hash functions in a kernel embed-

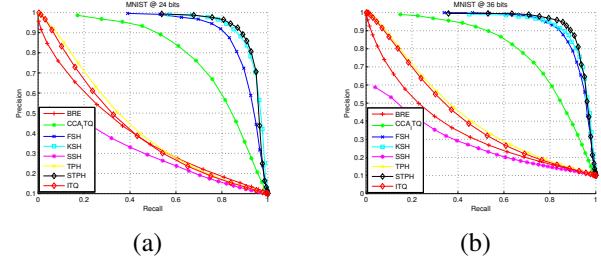


Figure 2. The precision recall curve on the MNIST dataset with regards to (a) 24 bits and (b) 36 bits.

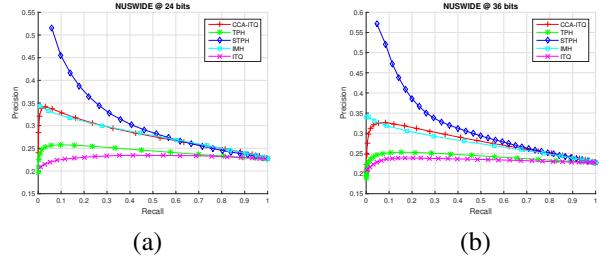


Figure 3. The precision recall curve on the NUSWIDE dataset with regards to (a) 24 bits and (b) 36 bits.

ding like us. This validates the effectiveness of the kernel feature. Another phenomenon that is worth noting is that, the metric precision@radius 2 often comes across a significant drop as the number of used bits increases, whereas metrics like mAP and precision@500 consistently increase. This is because the number of points falling in a bucket decrease exponentially when longer codes are used, leading to many failed queries by not returning any neighbor even in a Hamming ball of radius 2.

4. Conclusion

This paper has introduced a novel hashing technique called supervised topology preserving hashing (*STPH*) to learn compact binary codes for large scale visual search. *STPH* aims to leverage the label information and local

Table 3. Results on the NUSWIDE dataset with regards to different number of bits. The first two columns show the Hamming ranking results evaluated by mAP and precision@500. The right column shows the Hamming look-up results when the Hamming radius $r = 2$.

Method	mAP			precision@500			precision@r=2		
	12	24	36	12	24	36	12	24	36
CCA-ITQ	28.88	29.45	29.91	32.40	33.82	34.37	31.17	32.10	23.25
IMH	23.67	23.78	23.89	24.39	26.26	25.50	24.32	25.10	25.53
PCA-ITQ	25.85	25.89	25.87	27.57	29.25	29.60	25.25	26.92	27.84
TPH	24.73	24.73	24.61	28.40	30.23	29.92	26.65	28.26	11.14
STPH	29.75	29.86	31.19	26.26	29.44	31.48	30.16	31.28	31.03

topology structure of datasets to learn hash functions for better semantic neighbor search. The quantization error is explicitly minimized as a connection to the original and resultant hamming space. An efficient optimization process has been developed to minimize the resultant objective function that is composed of continuous and discrete variables. Experimental results show that STPH significantly improves the discriminability of the previous topology preserving hashing [21], and outperforms state-of-the-art hashing methods on three visual benchmarks, demonstrating its remarkable effectiveness for large-scale visual retrieval tasks.

Acknowledgements

This work is funded by the Youth Innovation Promotion Association, CAS (Grant No. 2015190) and the National Natural Science Foundation of China (Grant No. 61135002 and 61473289).

References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proc. of NIPS*, volume 14, pages 585–591, 2001. [2](#)
- [2] B. Chen, J. Wang, H. Zhao, N. Zheng, and J. Principe. Convergence of a fixed-point algorithm under maximum correntropy criterion. *Signal Processing Letters, IEEE*, 22(10):1723–1727, 2015. [2](#)
- [3] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proc. of ACM CIVR*, page 48, 2009. [3](#)
- [4] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *Proc. of VLDB*, volume 99, pages 518–529, 1999. [1](#)
- [5] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2916–2929, 2013. [1, 3](#)
- [6] R. He, Y. Cai, T. Tan, and L. Davis. Learning predictable binary codes for face indexing. *Pattern Recognition*, 2015. [1](#)
- [7] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, 2013. [2](#)
- [8] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Proc. of NIPS*, pages 1042–1050, 2009. [1, 3](#)
- [9] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proc. of IEEE CVPR*, pages 2130–2137, 2009. [1](#)
- [10] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2143–2157, 2009. [1](#)
- [11] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter. Fast supervised hashing with decision trees for high-dimensional data. In *Proc. of IEEE CVPR*, pages 1971–1978, 2014. [3](#)
- [12] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Proc. of IEEE CVPR*, pages 2074–2081, 2012. [1, 3](#)
- [13] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *Proc. of ICML*, pages 1–8, 2011. [1, 2](#)
- [14] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. [3](#)
- [15] F. Shen, C. Shen, W. Liu, and H. Shen. Supervised discrete hashing. In *Proc. of IEEE CVPR*, 2015. [1, 3](#)
- [16] F. Shen, C. Shen, Q. Shi, A. Van Den Hengel, and Z. Tang. Inductive hashing on manifolds. In *Proc. of IEEE CVPR*, pages 1562–1569, 2013. [3](#)
- [17] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):66–78, 2012. [1](#)
- [18] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008. [1](#)
- [19] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(12):2393–2406, 2012. [3](#)
- [20] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. of NIPS*, pages 1753–1760, 2009. [1, 2](#)
- [21] L. Zhang, Y. Zhang, X. Gu, J. Tang, and Q. Tian. Scalable similarity search with topology preserving hashing. *Image Processing, IEEE Transactions on*, 23(7):3025–3039, 2014. [1, 2, 3, 5](#)