APHASH: ANCHOR-BASED PROBABILITY HASHING FOR IMAGE RETRIEVAL

香港浸會大學 HONG KONG BAPTIST UNIVERSITY

Junjie Chen[1], Anran Wang[2], William K. Cheung[1]

[1]Hong Kong Baptist University

[2]Institute for Infocomm Research, A*STAR

Institute for Infocomm Research A*STAR

Abstract

We propose an unsupervised hashing method called Anchor-based Probability Hashing (i.e. APHash) to preserve the similarities by exploiting the distribution of data points:

- **Distances are transformed into probabilities in both original and hash spaces.**
- **Instead of constructing $n \times n$ probability matrices within the whole training set as in SePH[1], we first randomly select a small set of m anchors then construct asymmetric probability matrices of size $m \times n$ to avoid high complexity issue.**

Method

Step 1

In the original space, we construct probability matrix $\mathcal{P}$ between the small set of m anchors C and the whole training set X of n data items. Define $p_{j|i}$ as the probability of assigning x_j to anchor c_i. $\mathcal{P}$ is normalized row by row.

$$p_{j|i} = \begin{cases} 1, & \text{if} \quad d(\mathbf{c}_i, \mathbf{x}_j) \le \theta \\ 0, & \text{if} \quad d(\mathbf{c}_i, \mathbf{x}_j) > \theta \end{cases}$$

$d(c_i, x_j)$ denotes the Euclidean distance. θ is the threshold indicating the average distance between c_i and its k nearest neighbors computed as follows

$$\theta = \frac{\sum_{j \in \mathcal{N}_k(\mathbf{c}_i)} d(\mathbf{c}_i, \mathbf{x}_j)}{k}$$

Step 2

In hash space, we define $\mathcal{Q}$ as the probability distribution with Hamming distance. Inspired by t-SNE[2], we utilize t-distribution with one degree freedom to transform Hamming distance into probabilities.

$$q_{j|i} = \frac{(1 + g(\mathbf{h}_i, \mathbf{b}_j))^{-1}}{\sum_{t=1}^{n}(1 + g(\mathbf{h}_i, \mathbf{b}_t))^{-1}}$$

$\mathbf{h}_i$ and $\mathbf{b}_j$ denote hash codes of anchor point and training set item respectively. Hamming distance can be transformed to Euclidean distance with $g(\mathbf{h}_i, \mathbf{b}_j) = {}^1/_4 \|\mathbf{h}_i - \mathbf{b}_j\|_2^2$.

During optimization process, they are relaxed to real-value vectors $\hat{h}$ and $\hat{b}$ to make the problem tractable.

Step 3

- ❑ **The overall objective function of APHash containing two parts: KL-divergence loss and Quantization loss.**

$$J = J_0 + \lambda J_1$$

λ is a hyper parameter to balance two parts.

J_0: KL-divergence loss measures the difference between $\mathcal{P}$ and $\mathcal{Q}$ to make them as consistent as possible.

$$J_0 = \sum_{i=1}^{m}\sum_{j=1}^{n} p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

J_1: Quantization loss forces the relaxed entries of matrices $\widehat{H}$ and $\widehat{B}$ to be closed to ± 1 during optimization.

$$\mathcal{J}_1 = 1/Z_H \left\| |\widehat{H}| - 1 \right\|_2^2 + 1/Z_B \left\| |\widehat{B}| - 1 \right\|_2^2$$

- ❑ **We apply alternating stochastic gradient descent method to optimize the model.**
  - **We compute the derivative w.r.t. $\hat{h}$ and $\hat{b}$ as $\frac{\partial J}{\partial \hat{h}}$ and $\frac{\partial J}{\partial \hat{b}}$.**
  - **The overall objective is optimized w.r.t one parameter while fixing another until model converges.**
  - **we use $sign()$ function to obtain final hash code H and B**

Step 4

For out-of-sample extension, linear model is applied to learn hash function with the learned binary codes of anchor set H. The objective function is

$$L = \min_{W} \|H - W^T C\|_2^2 + \alpha \|W\|_2^2$$

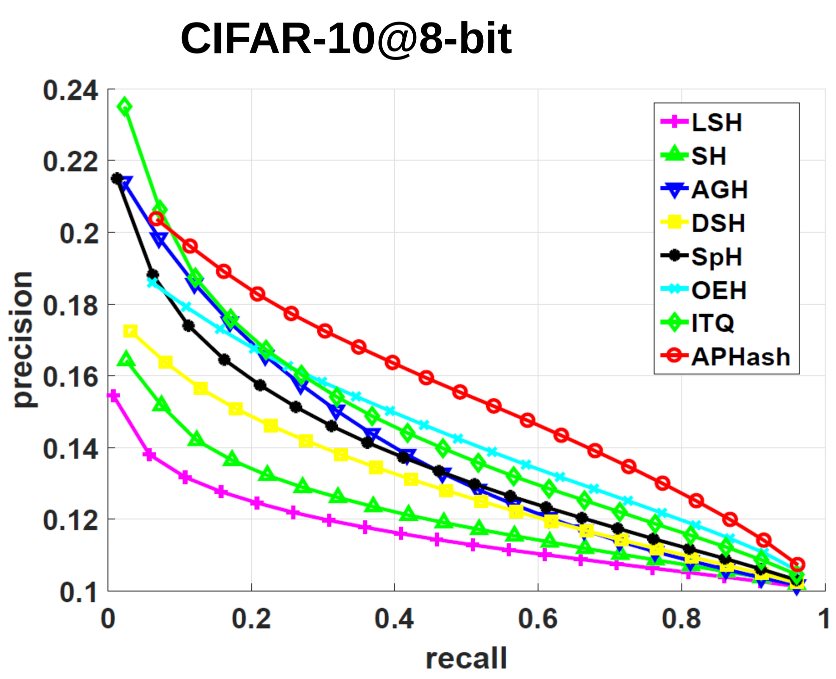
The learned binary code B is fixed and treated as index of database.

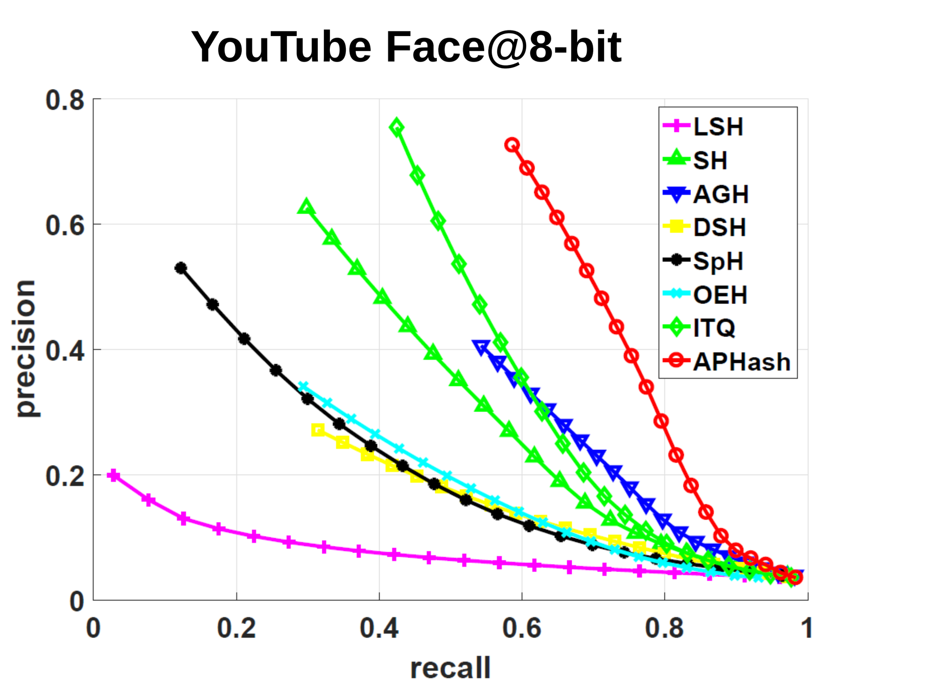
Experimental Results

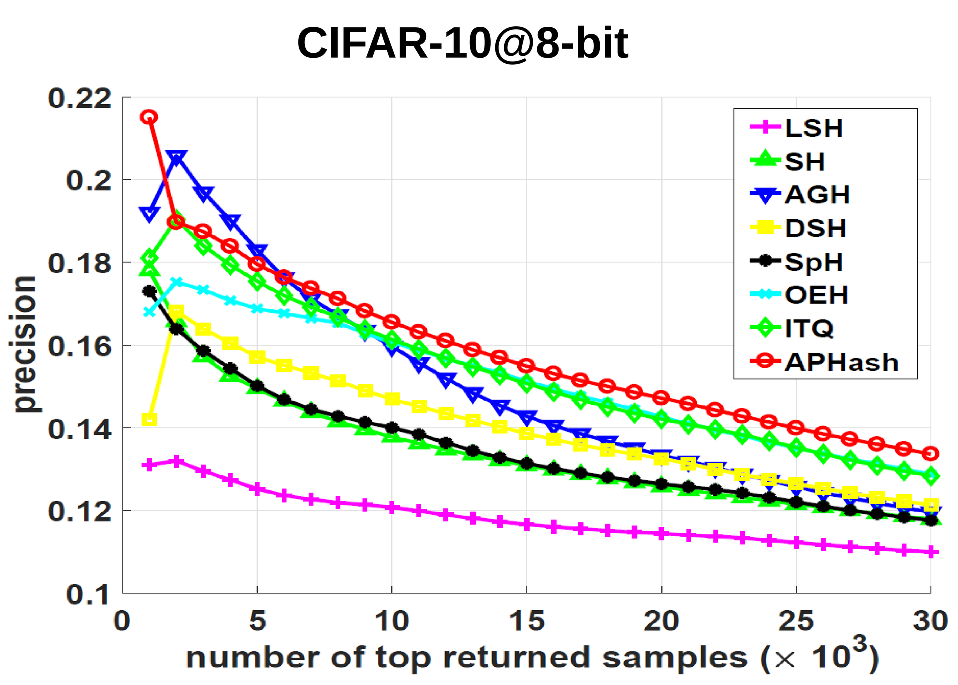
Two labeled datasets are used to evaluate the model: CIFAR-10 and YouTube Faces.

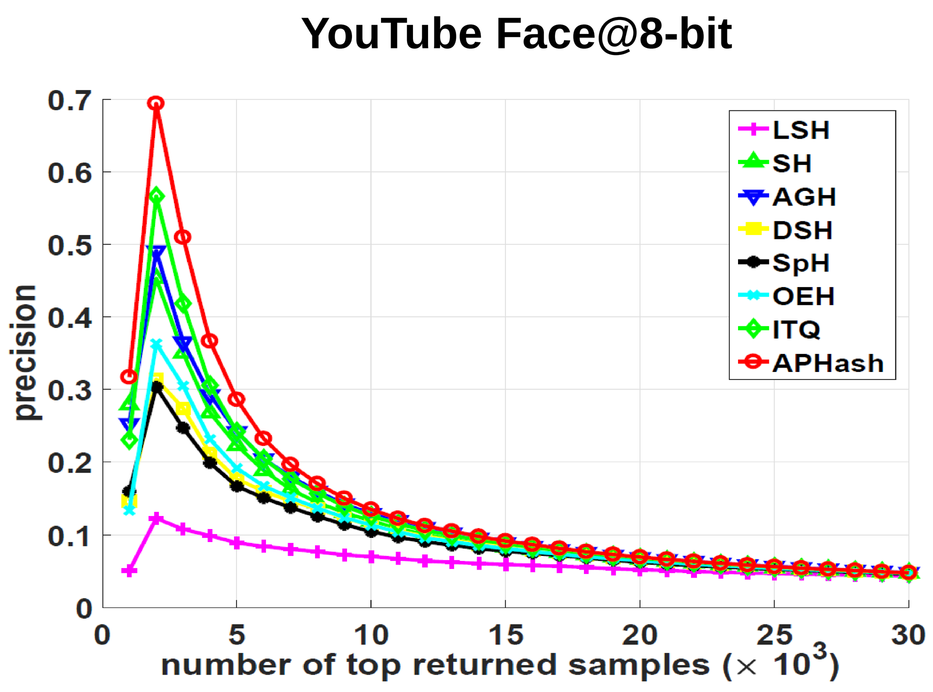
Table 1. Mean Average Precision of Hamming Ranking for different numbers of bits on two datasets.

Method	CIFAR-10 (mAP)				Youtube Faces (mAP)			
	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
LSH	0.1170	0.1222	0.1428	0.1515	0.1116	0.1586	0.2948	0.4354
SH	0.1295	0.1301	0.1303	0.1317	0.3900	0.5857	0.6652	0.5992
AGH	0.1507	0.1575	0.1483	0.1440	0.4527	0.6362	0.7299	0.6070
DSH	0.1470	0.1580	0.1625	0.1696	0.2754	0.3721	0.5207	0.5424
SpH	0.1465	0.1487	0.1537	0.1617	0.2646	0.3865	0.5030	0.5877
OEH	0.1373	0.1531	0.1572	0.1625	0.2182	0.4774	0.5901	0.6386
ITQ	0.1545	0.1650	0.1733	0.1787	0.4980	0.6709	0.7454	0.7525
APHash	**0.1630**	**0.1698**	**0.1779**	**0.1850**	**0.6160**	**0.6975**	**0.7499**	**0.7690**









Conclusions

- **We propose an unsupervised Anchor-based Probability Hashing, APHash.**
- **Basically, it learns informative hash codes by making use of the correlation between anchors and whole training set items.**
- **Experimental results on two datasets demonstrate the effectiveness of the proposed APHash.**

References

[1] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang, "Semantics-preserving Hashing for Cross-view Retrieval," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3864–3872.

[2] Laurens van der Maaten and Geoffrey Hinton, "Visualizing Data using t-sne," Journal of Machine Learning Research, vol. 9, no. Nov, pp. 2579 – 2605, 2008.