Isotropic Hashing

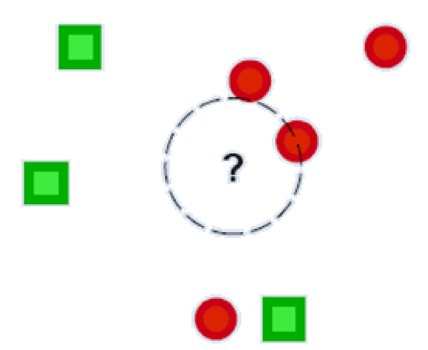
Weihao Kong
http://www.weihaokong.com
Shanghai Jiao Tong University

Wu-Jun Li
http://www.cs.sjtu.edu.cn/~liwujun/
Shanghai Jiao Tong University

Introduction

Nearest Neighbor Search

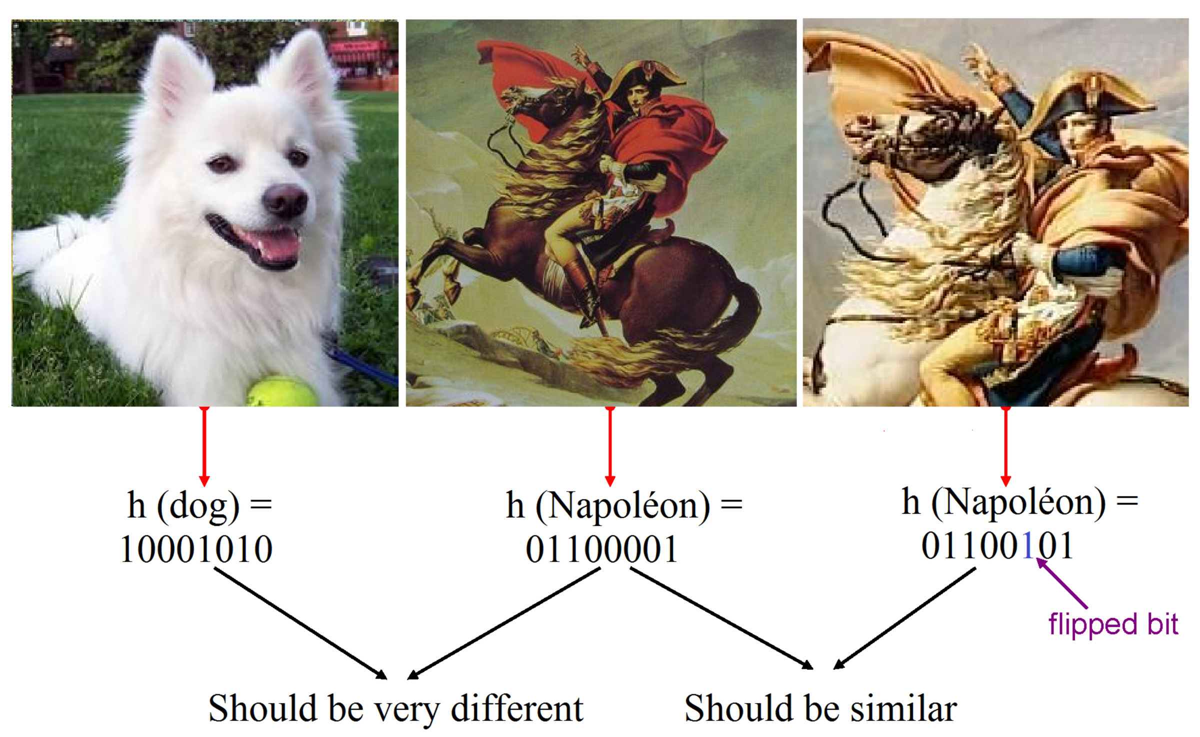
- Given a query point q, return the points closest to q
- Useful for finding similar objects in database.



- Brute force linear search is not practical in massive databases.

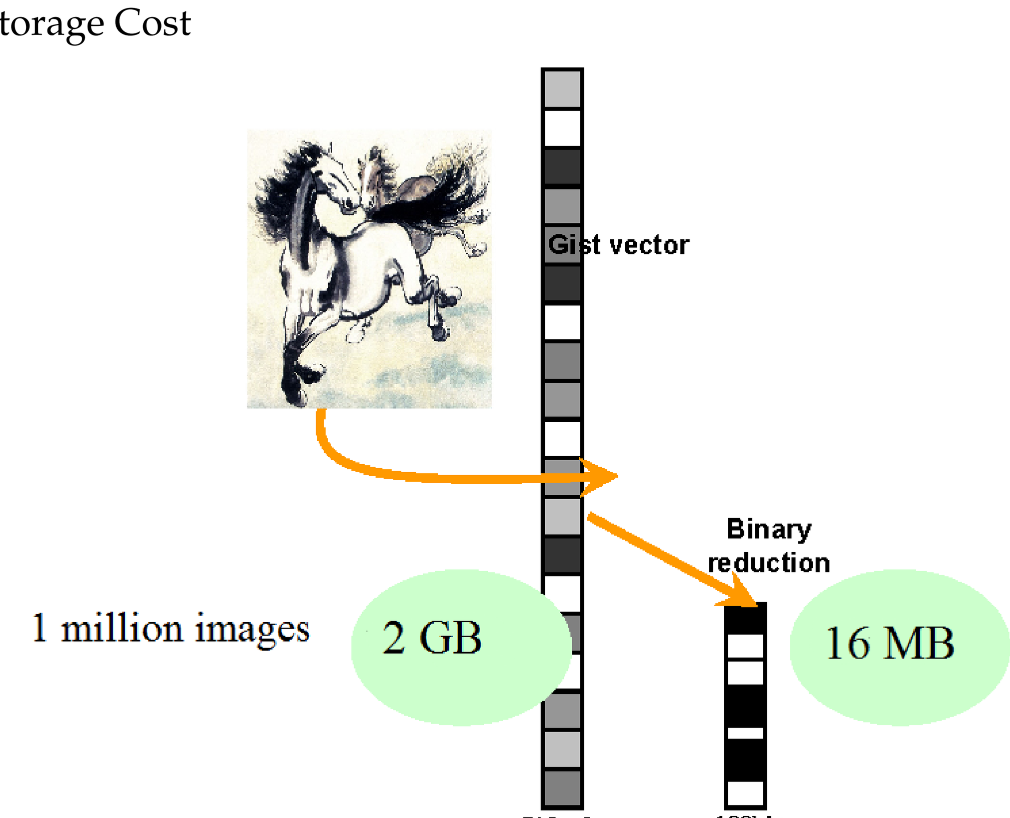
Applications: optical character recognition, k-nearest neighbor algorithm, content-based image retrieval, maximum likelihood decoding...

Hashing



Advantages of Hashing

- Fast Query Speed
  - By using hashing scheme, we can achieve constant or sub-linear search time complexity.
  - Exhaustive search is also acceptable because the comparison cost is cheap now.
- Reduce Storage Cost



PCAH

To generate a code of m bits, PCAH performs PCA on X, and then use the top m eigenvectors of the matrix XX^T as columns of the projection matrix $W \in \mathbb{R}^{d\times m}$. Here, top m eigenvectors are those corresponding to the m largest eigenvalues $\{\lambda_k\}_{k=1}^m$, generally arranged with the non-increasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Let $\lambda = [\lambda_1, \lambda_2, \cdots, \lambda_m]^T$. Then

$$\Lambda = W^T X X^T W = \text{diag}(\lambda)$$

Define hash function

$$h(\mathbf{x}) = sgn(W^T \mathbf{x})$$

Isotropic Hashing

the idea of our IsoHash method is to learn an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ which makes $Q^T W^T X X^T W Q$ become a matrix with equal diagonal values. The effect of the orthogonal matrix Q is to make each projected dimension has the same variance while keeping the Euclidean distances between any two points unchanged.

Let

$$\mathbf{a} = [a_1, a_2, \cdots, a_m] \text{ with } a_i = a = \frac{\sum_{i=1}^m \lambda_i}{m}, \tag{1}$$

and

$$\mathcal{T}(\mathbf{z}) = \{T \in \mathbb{R}^{m\times m} | \text{diag}(T) = \text{diag}(\mathbf{z})\},$$
$$\mathcal{M}(\Lambda) = \{Q^T \Lambda Q | Q \in \mathcal{O}(m)\}, \tag{2}$$

where $\mathcal{O}(m)$ is the set of all orthogonal matrices in $\mathbb{R}^{m\times m}$.

Problem 1. *The **problem of IsoHash** is to find an orthogonal matrix Q making $Q^T W^T X X^T W Q \in \mathcal{T}(\mathbf{a})$, where $\mathbf{a}$ is defined in (1).*

Lemma 1. *[Schur-Horn Lemma [3]] Let $\mathbf{c} = \{c_i\} \in \mathbb{R}^m$ and $\mathbf{b} = \{b_i\} \in \mathbb{R}^m$ be real vectors in non-increasing order respectively, i.e., $c_1 \geq c_2 \geq \cdots \geq c_m$, $b_1 \geq b_2 \geq \cdots \geq b_m$. There exists a Hermitian matrix H with eigenvalues $\mathbf{c}$ and diagonal values $\mathbf{b}$ if and only if*

$$\sum_{i=1}^k b_i \leq \sum_{i=1}^k c_i, \text{ for any } k = 1, 2, ..., m,$$
$$\sum_{i=1}^m b_i = \sum_{i=1}^m c_i.$$

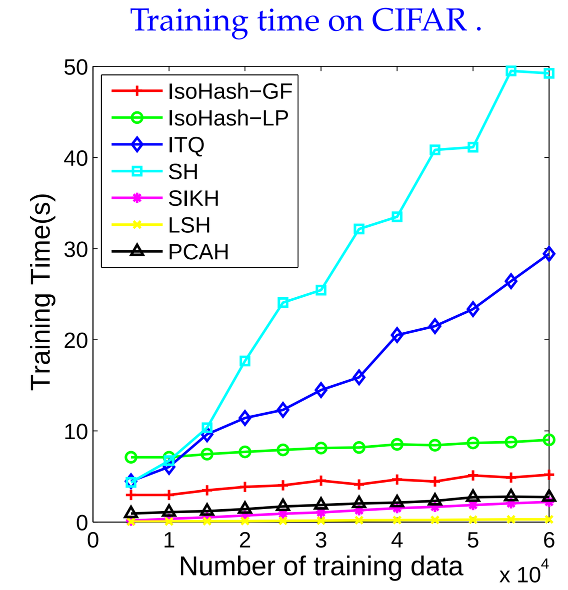
Corollary 1. *There exists a solution to the IsoHash problem. And this solution is in the intersection of $\mathcal{T}(\mathbf{a})$ and $\mathcal{M}(\Lambda)$.*

Experiment

mAP on LabelMe and CIFAR data sets

Method	LabelMe					CIFAR				
# bits	32	64	96	128	256	32	64	96	128	256
IsoHash-GF	0.2580	0.3269	0.3528	0.3662	0.3889	0.2249	0.2969	**0.3256**	**0.3357**	0.3600
IsoHash-LP	0.2534	0.3223	**0.3577**	**0.3826**	0.4274	0.1907	0.2624	0.3027	0.3223	**0.3651**
PCAH	0.0516	0.0401	0.0341	0.0307	0.0232	0.0319	0.0274	0.0241	0.0216	0.0168
ITQ	**0.2786**	**0.3328**	0.3504	0.3615	0.3728	**0.2490**	**0.3051**	0.3238	0.3319	0.3436
SH	0.0826	0.1034	0.1447	0.1653	0.2080	0.0510	0.0589	0.0802	0.1121	0.1535
SIKH	0.0590	0.1482	0.2074	0.2526	**0.4488**	0.0353	0.0902	0.1245	0.1909	0.3614
LSH	0.1549	0.2574	0.3147	0.3375	0.4034	0.1052	0.1907	0.2396	0.2776	0.3432

Training time on CIFAR .



Lift and Projection (LP)

- Lift step:
  Given a $T^{(k)} \in \mathcal{T}(\mathbf{a})$, we find the point $Z^{(k)} \in \mathcal{M}(\Lambda)$ such that $||T^{(k)} - Z^{(k)}||_F = dist(T^{(k)}, \mathcal{M}(\Lambda))$, where $dist(T^{(k)}, \mathcal{M}(\Lambda))$ denotes the minimum distance between $T^{(k)}$ and the points in $\mathcal{M}(\Lambda)$.
- Projection step:
  Given a $Z^{(k)}$, we find $T^{(k+1)} \in \mathcal{T}(\mathbf{a})$ such that $||T^{(k+1)} - Z^{(k)}||_F = dist(\mathcal{T}(\mathbf{a}), Z^{(k)})$, where $dist(\mathcal{T}(\mathbf{a}), Z^{(k)})$ denotes the minimum distance between $Z^{(k)}$ and the points in $\mathcal{T}(\mathbf{a})$.

The projection operation is easy to complete. For the lift operation, we have the following Theorem 1 [1].

Theorem 1. *Suppose $T = Q^T D Q$ is an eigen-decomposition of T where $D = diag(\mathbf{d})$ with $\mathbf{d} = [d_1, d_2, ..., d_m]^T$ being T's eigenvalues which are ordered as $d_1 \geq d_2 \geq \cdots \geq d_m$. Then the nearest neighbor of T in $\mathcal{M}(\Lambda)$ is given by*

$$Z = Q^T \Lambda Q. \tag{3}$$

Algorithm 1 Lift and projection based IsoHash (IsoHash-LP)

Input: $X \in \mathbb{R}^{d\times n}, m \in \mathbb{N}^+, t \in \mathbb{N}^+$

- $[\Lambda, W] = PCA(X, m)$.
- Generate a random orthogonal matrix $Q_0 \in \mathbb{R}^{m\times m}$.
- $Z^{(0)} \leftarrow Q_0^T \Lambda Q_0$.
- **for** $k = 1 \to t$ **do**
  Calculate $T^{(k)}$ from $Z^{(k-1)}$
  Perform eigen-decomposition of $T^{(k)}$ to get $Q_k^T D Q_k = T^{(k)}$.
  Calculate $Z^{(k)}$ from Q_k and Λ by equation (3).
- **end for**
- $Y = sgn(Q_t^T W^T X)$.

Output: Y

Hashing Method

- **ITQ:** ITQ uses an iteration method to find an orthogonal rotation matrix to refine the initial projection matrix learned by PCA so that the quantization error of mapping the data to the vertices of binary hypercube is minimized. Experimental results in [2] show that it can achieve better performance than most state-of-the-art methods.
- **SH:** SH uses the eigenfunctions computed from the data similarity graph for projection [5].
- **SIKH:** SIKH uses random projections to approximate the shift-invariant kernels. As in [2, 4], we use a Gaussian kernel whose bandwidth is set to the average distance to the 50th nearest neighbor.
- **LSH:** LSH uses a Gaussian random matrix to perform random projection.

Gradient Flow

The objective function can be reformulated as follows [?]:

$$\min_{Q\in\mathcal{O}(m)} F(Q) = \frac{1}{2}||\text{diag}(Q^T \Lambda Q) - \text{diag}(\mathbf{a})||_F^2. \tag{4}$$

The gradient ∇F at Q can be calculated as

$$\nabla F(Q) = 2\Lambda\beta(Q), \tag{5}$$

where $\beta(Q) = \text{diag}(Q^T \Lambda Q) - \text{diag}(\mathbf{a})$. Once we have computed the gradient of F, it can be projected onto the manifold $\mathcal{O}(m)$ according to the following Theorem 2 [1]..

Theorem 2. *The projection of $\nabla F(Q)$ onto $\mathcal{O}(m)$ is given by*

$$g(Q) = Q[Q^T \Lambda Q, \beta(Q)] \tag{6}$$

where $[A, B] = AB - BA$ is the Lie bracket.

The vector field $\dot{Q} = -g(Q)$ defines a steepest descent flow on the manifold $\mathcal{O}(m)$ for function $F(Q)$. Letting $Z = Q^T \Lambda Q$ and $\alpha(Z) = \beta(Q)$, we get

$$\dot{Z} = [Z, [\alpha(Z), Z]], \tag{7}$$

where $\dot{Z}$ is an isospectral flow that moves to reduce the objective function $F(Q)$.

Algorithm 2 Gradient flow based IsoHash (IsoHash-GF)

Input: $X \in \mathbb{R}^{d\times n}, m \in \mathbb{N}^+$

- $[\Lambda, W] = PCA(X, m)$.
- Generate a random orthogonal matrix $Q_0 \in \mathbb{R}^{m\times m}$.
- $Z^{(0)} \leftarrow Q_0^T \Lambda Q_0$.
- Start integration from $Z = Z^{(0)}$ with gradient computed from equation (7).
- Stop integration when reaching a stable equilibrium point.
- Perform eigen-decomposition of Z to get $Q^T \Lambda Q = Z$.
- $Y = sgn(Q^T W^T X)$.

Output: Y

References

References

[1] M. Chu and K. Driessel. The projected gradient method for least squares matrix approximations with spectral constraints. *SIAM Journal on Numerical Analysis*, pages 1050–1060, 1990.

[2] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of Computer Vision and Pattern Recognition*, 2011.

[3] A. Horn. Doubly stochastic matrices and the diagonal of a rotation matrix. *American Journal of Mathematics*, 76(3):620–630, 1954.

[4] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Proceedings of Neural Information Processing Systems*, 2009.

[5] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proceedings of Neural Information Processing Systems*, 2008.