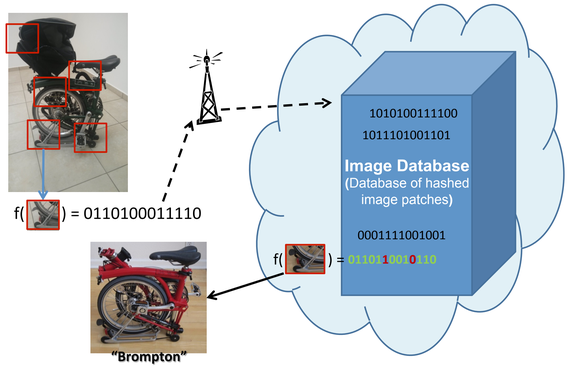
The Power of Asymmetry in Binary Hashing

Behnam Neyshabur TTI-Chicago
Payman Yadollahpour TTI-Chicago
Yury Makarychev TTI-Chicago
Ruslan Salakhutdinov University of Toronto
Nathan Srebro TTI-Chicago, Technion

Binary Hashing

Hash function $f : \mathcal{X} \to \{\pm 1\}^k$ that preserves a given similarity relation between objects:

$$[d_{\text{hamming}}(f(x), f(x')) < \theta] \approx \text{sim}(x, x')$$



The Power of Asymmetry

Even if $\text{sim}(x, x')$ is symmetric and "well behaved", we can gain by using two different hash functions:

- Use $f(x)$ to hash each **query**
- Use $g(x)$ to hash objects in **database**

$$[d_{\text{hamming}}(f(x), g(x')) < \theta] \approx \text{sim}(x, x')$$

[Dong et al. SIGIR 08] and [Gordo et al. CVPR 11] also discuss "asymmetric hashing", in that real-valued queries are compared to binary hashes in database. They use:

$$f : \mathcal{X} \to \mathbb{R}$$
$$g(x) = \text{sign}(f(x))$$

That is, the two mappings differ only in their precision. Here, we propose using two entirely different binary hash functions.

Arbitrary Binary Codes

We are given a similarity $S_{ij} = \text{sim}(x_i, x_j)$ over objects $\mathcal{X} = \{x_1, \ldots, x_n\}$ and want mapping $f : \mathcal{X} \to \{\pm 1\}^k$ s.t.

$$\forall_{ij}\ S_{ij} = [d(f(x_i), f(x_j)) < \theta]$$

mapping $f(\cdot)$ can be specified by $u_i = f(x_i)$

- What is shortest possible bit length k?

$$\exists u_1, \ldots, u_n \in \{\pm 1\}^k, \theta \in \mathbb{R} \quad \forall_{ij}\ S_{ij} = [d(u_i, u_j) < \theta]$$

- What is shortest bit length k if we allow asymmetry? $v_j = g(x_j)$

$$\exists u_1, \ldots, u_n, v_1, \ldots, v_n \in \{\pm 1\}^k, \theta \in \mathbb{R} \quad \forall_{ij}\ S_{ij} = [d(u_i, v_j) < \theta]$$

Formulation as Matrix Factorization

Symmetric	Asymmetric
$\min k$ s.t $\text{sign}(U^\top U - \theta) = S$	$\min k$ s.t $\text{sign}(U^\top V - \theta) = S$
$U \in \{\pm 1\}^{k \times n}$	$U, V \in \{\pm 1\}^{k \times n}$

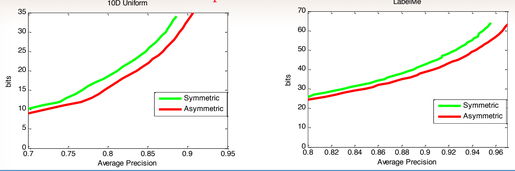
The Power of Asymmetry for Arbitrary Binary Codes

Theorem: For any r, there exists a set of points in Euclidean space, such that for the similarity

$$S(x_i, x_j) = [\|x_i - x_j\|_2 < 1]$$

using symmetric binary hash, we need $k \geq 2^r$ bits, using asymmetric binary hash, we need $k \leq 2r$ bits to exactly capture the similarity relation.

Empirical Evaluation



Proof sketch

$$X^\top X = \frac{1}{n} \begin{bmatrix} n & -1 & \cdots & -1 & 1 & 1 & \cdots & 1 \\ -1 & n & \cdots & -1 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & n & 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 & n & -1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 & -1 & n & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & -1 & -1 & \cdots & n \end{bmatrix}, \quad n = 2^r$$

Asymmetric:

- Let B be an $r \times n$ matrix whose column vectors are vertices of the cube $\{\pm 1\}^n$ (in any order).
- Let C be an $r \times n$ matrix whose entries in the first $\frac{n}{2}$ columns are 1 and the remaining entries are -1.
- If $U = \begin{bmatrix} B \\ C \end{bmatrix}, V = \begin{bmatrix} B \\ -C \end{bmatrix}$ and $\theta = -1$, then $S = \text{sign}(U^\top V - \theta)$

Symmetric:

- Let $Y' = U^\top U$
- $Y'_{ij} \in [-k_s, k_s]$ and thus $\theta \in [-k_s + 1, k_s - 1]$
- Let $q = [1, \ldots, 1, -1, \ldots, -1]^\top$
- We have:

$$\begin{aligned} 0 &\leq q^\top Y' q \\ &= \sum_{i=1}^{n} Y'_{ii} + \sum_{i,j:S_{ij}=-1} Y'_{ij} - \sum_{i,j:S_{ij}=1, i \neq j} Y'_{ij} \\ &\leq \sum_{i=1}^{n} k_s + \sum_{i,j:S_{ij}=-1} (\theta - 1) - \sum_{i,j:S_{ij}=1, i \neq j} (\theta + 1) \\ &= nk_s + (0.5n^2 - n)(\theta - 1) - 0.5n^2(\theta + 1) \\ &\leq 2nk_s - n^2 \end{aligned}$$

Parametric Mappings

- Use $f(x) = \phi_F(x)$ in some (typically parametric) family $\phi_F \in \mathcal{F}$
  - E.g. $\phi_F(x) = \text{sign}(Fx), F \in \mathbb{R}^{k \times d}$
  - Could be more complex, e.g. multi-layer network, kernel-based, etc.
- Why restrict $f = \phi_F \in \mathcal{F}$?
  - Generalization: learn F using objects $x_1, \ldots, x_n$, then receive new objects x as queries
  - Compactness of representation
- Asymmetric extension

$$\text{sim}(x_i, x_j) \approx [d(\phi_F(x_i), \phi_G(x_i)) < \theta]$$

  - Learn params F, G from training data (=pairs of database objects)
  - Hash queries using ϕ_F
  - Hash database objects using ϕ_G

Optimization

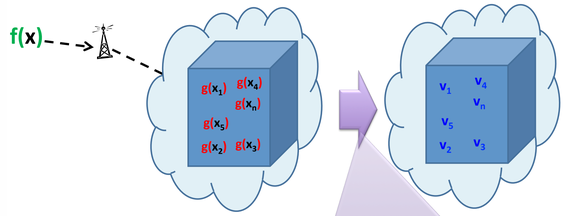
Local search in highly non-convex problem with many discontinuities.

Main observation: Optimizing a single row of F entails solving a single weighted binary classification problem.

$f(x) = \text{sign}(Fx)$
$g(x) = \text{sign}(Gx)$

Alternating updates:
- Fix F and update rows of G
- Fix G and update rows of F

Even More Power



✓ No need to generalize: only used on database objects
✓ Stored in database hash, no need for compactness
✓ Can use arbitrary mapping $g(x_i) = v_i$

- Learn parametric ϕ_F and arbitrary vectors $v_1, \ldots, v_n$ such that on $x_1, \ldots, x_n$

$$\text{sim}(x_i, x_j) \approx [d(\phi_F(x_i), v_j) < \theta]$$

- For query x, use $d(\phi_F(x), v_j)$ to approximate $\text{sim}(x, x_j)$ e.g. find v_j in database with small $d(\phi_F(x), v_j)$

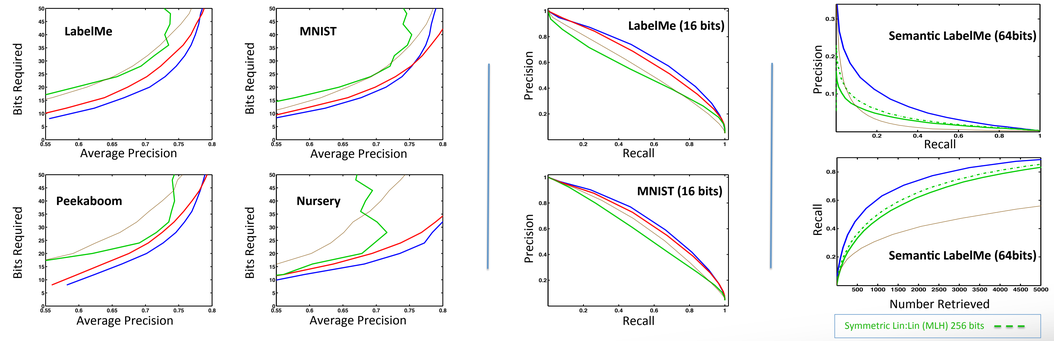
Empirical Results

Asymmetric Lin:Arb

Asymmetric Lin:Lin

Symmetric Lin:Lin (MLH) [Norouzi et al. ICML 2011]

Symmetric Kernel-based (KSH) [Liu et al. CVPR 2012]



References

[Norouzi et al. ICML 2011] M. Norouzi and D. J. Fleet. Minimal loss hashing for compact binary codes. *ICML*, 2011.
[Liu et al. CVPR 2012] W. Liu, R. Ji J. Wang, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. CVPR, 2012
[Gordo et al. CVPR 11] A. Gordo and F. Perronnin. Asymmetric distances for binary embeddings. *CVPR*, 2011.
[Dong et al. SIGIR 08] W. Dong and M. Charikar. Asymmetric distance estimation with sketches for similarity search in high- dimensional spaces. *SIGIR*, 2008.

Asymmetric hashes can enable better approximation with shorter bit-length!