Compact Kernel Hashing with Multiple Features

Xianglong Liu, Junfeng He†, Di Liu‡, Bo Lang**

*State Key Lab of Software Development Environment, Beihang University, Beijing, China

†Columbia University, New York, NY 10027, U.S.A.

‡China Academy of Telecommunication Research of MIIT, Beijing, China

Background and Related Work

Background

- The explosive growth of the vision data motivates the recent studies on hashing based nearest neighbor search (ANN)
- Locality-Sensitive Hashing (LSH) gives the paradigm of hashing based ANN
- Various scenarios: unsupervised, supervised, kernelized, and multiple probes

Related Work

- Adaptively combining diverse complementary features can give improved performance
- Existing multiple feature hashing approaches either simply post-combine linear outputs of each feature type or equally pre-concatenate all features as one

Main Issues

- The correlation and importance of each feature type are still not fully exploited
- Computationally expensive in both training and searching

Multiple Feature Kernel Hashing

Notations

- a set of N training examples with M visual features
  - $X_n^{(m)} \in R^{d_m \times 1}$: the m-th feature (d_m dimension) of n-th sample
  - $X^{(m)} = \left[X_1^{(m)}, X_2^{(m)}, \dots, X_n^{(m)}\right]$: the m-th feature of all data

Key Idea

- Learn a kernel space incorporating multiple features, where the neighbor relationships can be well preserved.

Nonlinear feature mapping

- a series of embedding functions $\phi_m(\cdot)$ corresponding to each visual feature
- nonlinear mapping of i-th sample

$$\phi(X_i) = \left[\mu_1^{\frac{1}{2}}\phi_1^T\left(X_i^{(1)}\right), \dots, \mu_M^{\frac{1}{2}}\phi_M^T\left(X_i^{(M)}\right)\right]^T$$

- linear projection hashing

$$h_p(X_i) = sign\left(V_p^T\phi(X_i) + b_p\right),\quad p = 1,\dots,P.$$

Multiple kernel form

- V_p in kernel space can be represented as a combination of R landmarks Z_r

$$V_p = \sum_{r=1}^{R} W_{rp}\phi(Z_r), r = 1,\dots,R.$$

- let $K^{(m)}$ denote the kernel corresponding to $\phi_m(\cdot)$, then $\phi(\cdot)$ defines a kernel $K = \sum_{m=1}^{M}\mu_m K^{(m)}$
- kernel hashing

$$h_p(X_i) = sign\left(W_{\cdot p} K_i + b_p\right),\quad p = 1,\dots,P.$$

Optimization

Optimization

- Objective function similar to that of spectral hashing

$$\mathcal{L}(S, W, b, \mu) = \frac{1}{2}\boxed{\sum_{i,j=1}^{N} S_{ij}\|Y_i - Y_j\|^2} + \lambda\boxed{\|V\|_F^2}$$

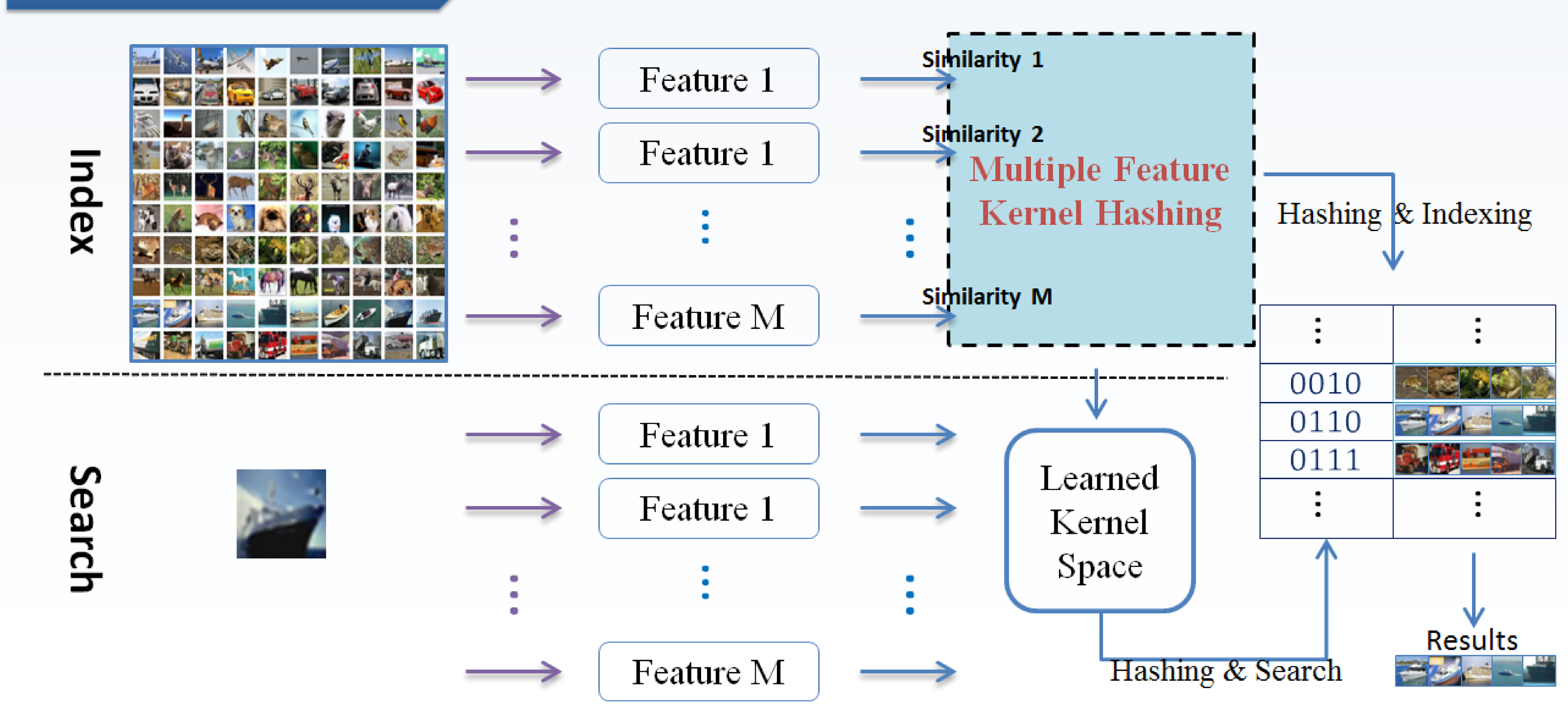
Spectral embedding loss regularization

- Alternating optimization
  - Update hashing parameters: (W, b)
  - Update linear combination coefficients: μ

methods	training	search
CHMS	$O(T(D^3 + D^2N + DN_s))$	$O(PD)$
MFH	$O(D^3 + D^2N + DN_s)$	$O(PD)$
Proposed	$O(T(LN_s + P^3))$	$O(PM + ML)$

[1] Here $P, M, T \ll D, N_s \ll N^2$ and $L < D$ ($D = \sum_{m=1}^{M} d_m$).

Framework

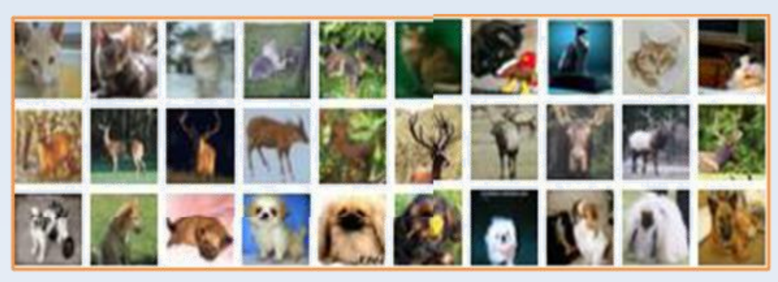


Experiments

Datasets

- CIFAR-10: 60K, 384D GIST + 300D SIFT BoW
- NUS-WIDE: 270K, 128D texture + 225D color

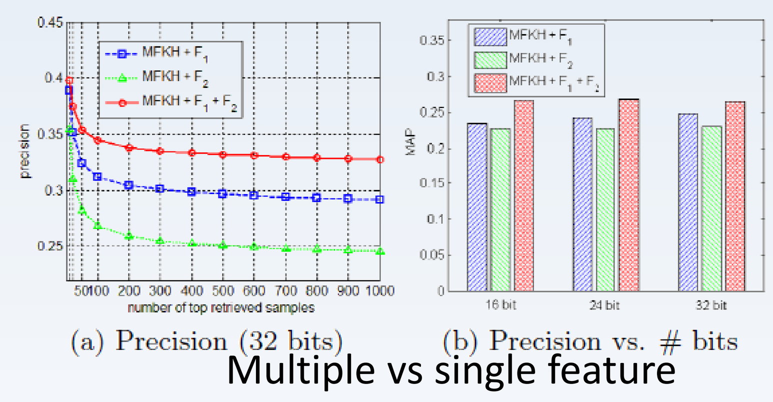
CIFAR-10



NUS-WIDE

Measurement

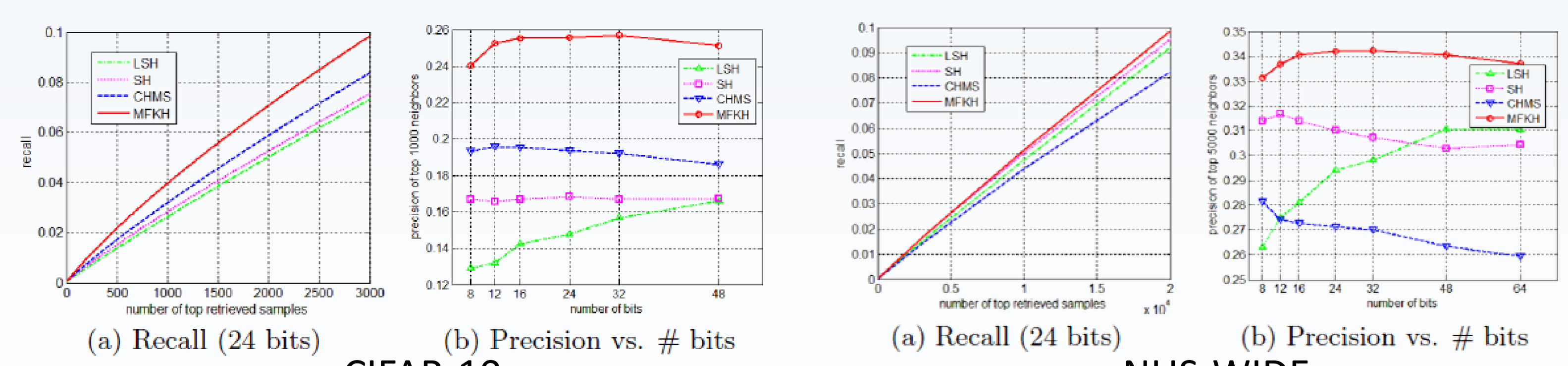
- MAP for Hamming ranking
- Recall, precision of top results



(a) Precision (32 bits) (b) Precision vs. # bits

Multiple vs single feature

Results



(a) Recall (24 bits) (b) Precision vs. # bits (a) Recall (24 bits) (b) Precision vs. # bits

CIFAR-10 NUS-WIDE

Conclusion

- Efficient multiple feature hashing.
  - similarity preserving hashing with linearly combined multiple kernels
  - efficient alternating optimizing way