检索与推荐中的排序和索引问题

程　健　研究员

jcheng@nlpr.ia.ac.cn

中国科学院自动化研究所

大 纲

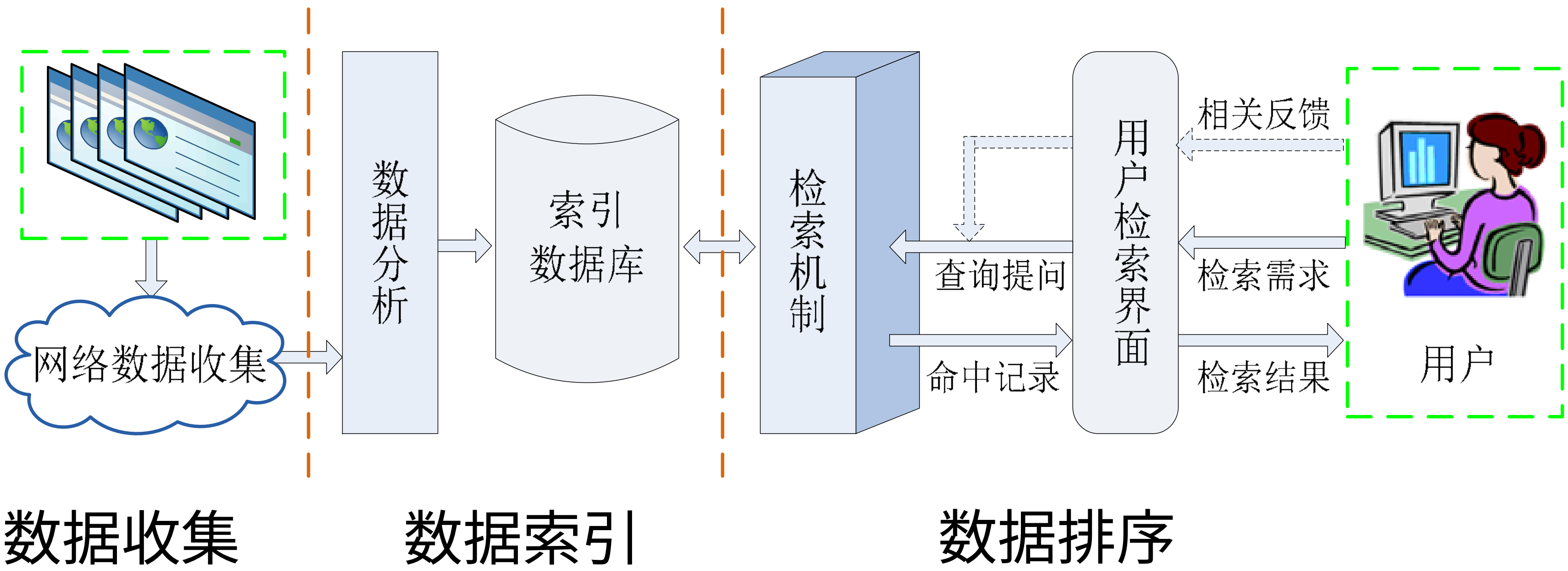
- 背景介绍
- 排序与索引
- 近似近邻搜索
- 总结与展望

多媒体内容索引与排序

浩如烟海
无从下手

唯有索引
方得真经

网络多媒体检索的基本流程



Performance Evaluation

- Measures for top-N recommendation
  - NDCG(Normalized Discounted Cumulative Gain)

$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}$$

定义不唯一

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

Ideal DCG

大 纲

- 背景介绍
- 排序与索引
- 近似近邻搜索
- 总结与展望

多媒体内容索引

- 排序准则 = 动态相关性 +静态相关性
  - 动态相关性：与查询词的相关性
    - TF, IDF
    - Title, Body, Anchor, URL
    - Proximity
    - …

TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

  - 静态相关性：查询对象集的质量（权威性）
    - PageRank
    - PageQuality, Spam
    - …

与查询词的相关性

- 查询词在文档中出现的位置：前 > 后
- 出现在Tag - “Title and Description”
- 出现在Tag - “Keyword”
  - 注意：**Spam** 问题
    - If a query term appears in this tag,
      - It must appear somewhere in the body, otherwise penalize
      - Query term should not appear more than twice in this tag, otherwise penalize
- TF-IDF
- 文件大小:
  - < 40K is preferred, very large file is penalized
- ……

PageRank

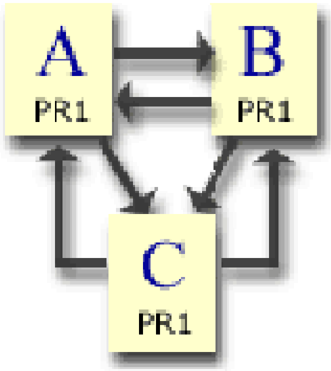
- PageRank的基本公式

$$r(p) = \alpha \times \sum_{q:(q,p)\in\varepsilon} \frac{r(q)}{\omega(q)} + (1-\alpha) \times \frac{1}{N}$$

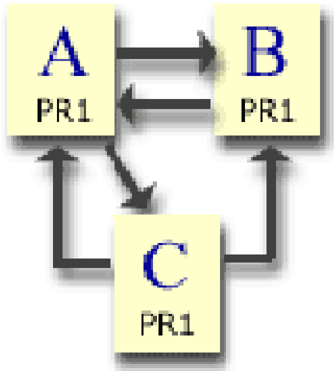
  - *q* – 网页 *p*的后向链接, i.e., *q* 指向 *p*
  - *ω(q)*- 网页 *q*的前向链接数目.
  - *r(q)* – 网页 *q*的PageRank.
  - *N* - 整个网络中网页的总数

S. Brin, L. Page: The Anatomy of a Large-scale Hypertextual Web Search Engine Computer Networks and ISDN Systems. WWW 1998

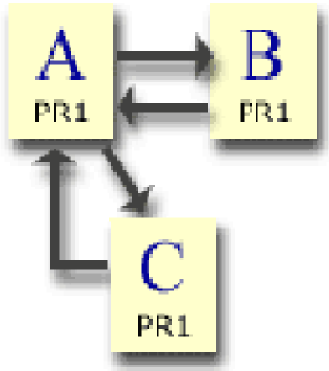
PageRank



Page A = 1
Page B = 1
Page C = 1



Page A = 1.425
Page B = 1
Page C = 0.575



Page A = 1.4594
Page B = 0.7702
Page C = 0.7702

PageRank

优点：与查询无关的静态算法；

缺点：与主题无关，旧网页比新网页排名高

S. Brin, L. Page: The Anatomy of a Large-scale Hypertextual Web Search Engine Computer Networks and ISDN Systems. WWW 1998

HITS

基本假设:

✓一个好的“ Authority”页面会被很多好的“ Hub”页面指向；

✓ 一个好的“ Hub”页面会指向很多好的“ Authority”页面；

优点：自然语言、社交网络取得很好效果；

缺点： 计算复杂、主题漂移、易作弊

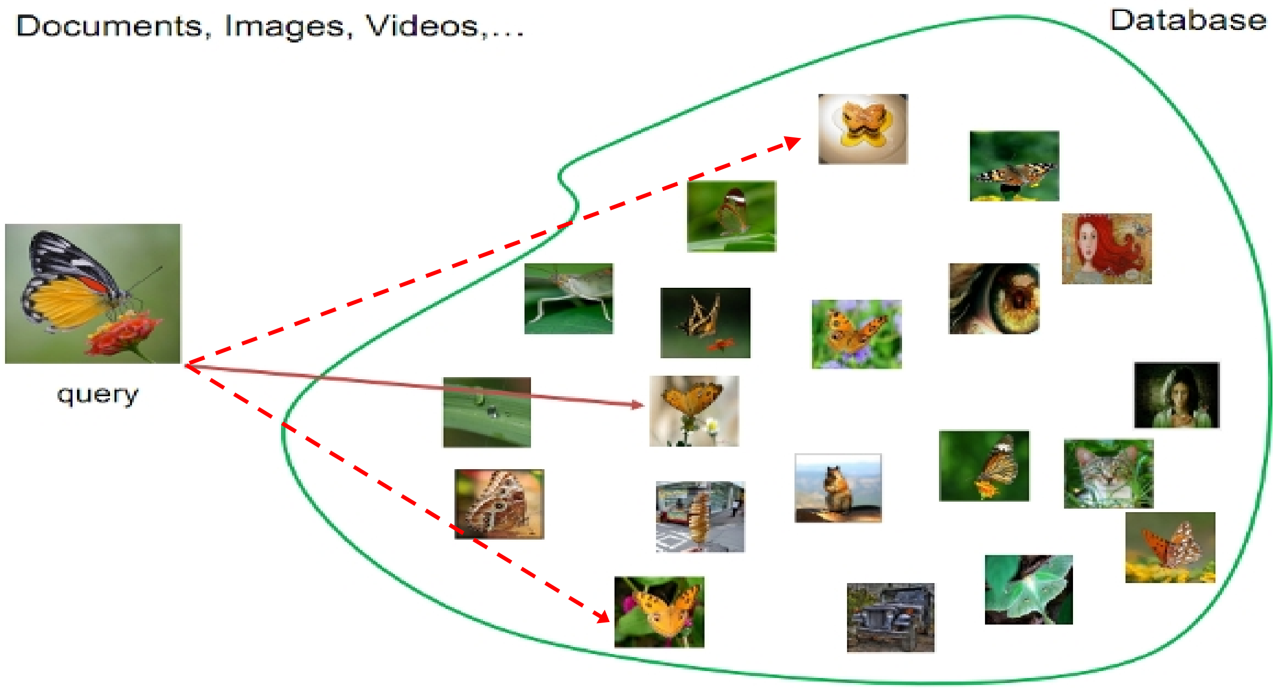
J.M.Kleinberg. Authoritative sources in a hyperlinked environment. Annual ACM-SIAM Symposium on Discrete Algorithms, 1998

大 纲

- 背景介绍
- 排序与索引
- 近似近邻搜索
- 总结与展望

近似近邻搜索

- 近似近邻：检索精度和检索时间的平衡
- 对大多数应用，近似近邻足够满足需求



基于哈希的表示

- **How to index data?**

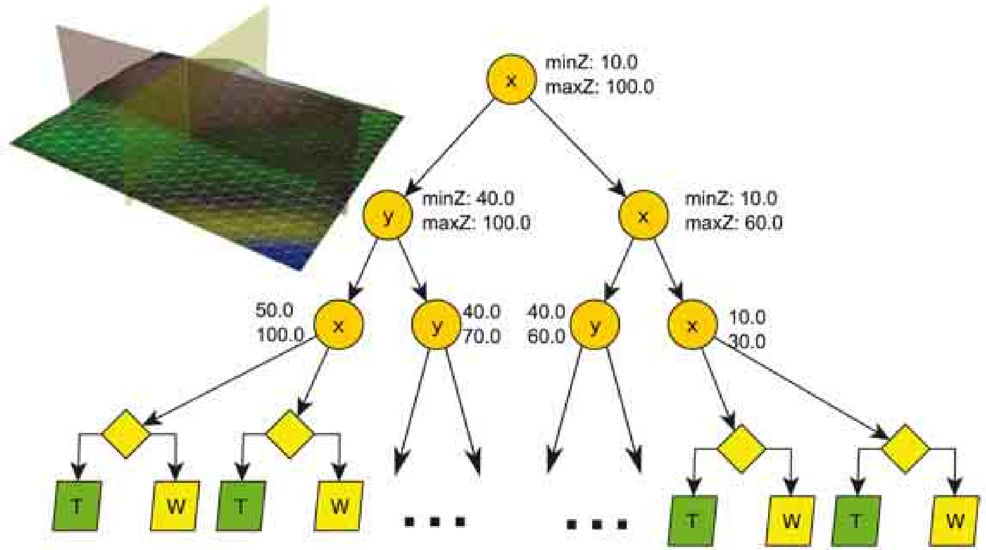
TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Document analysis

Tree-based

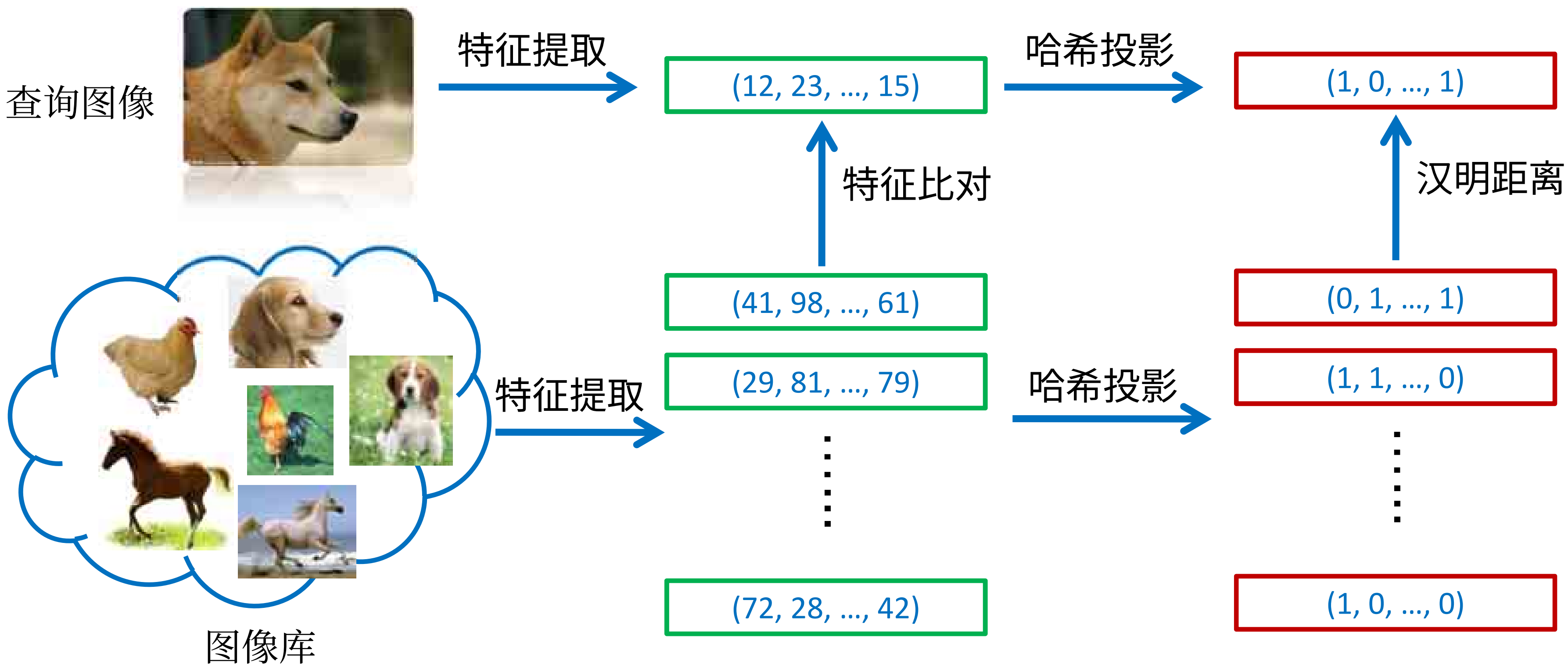


Curse of dimensionality

Pictures from web

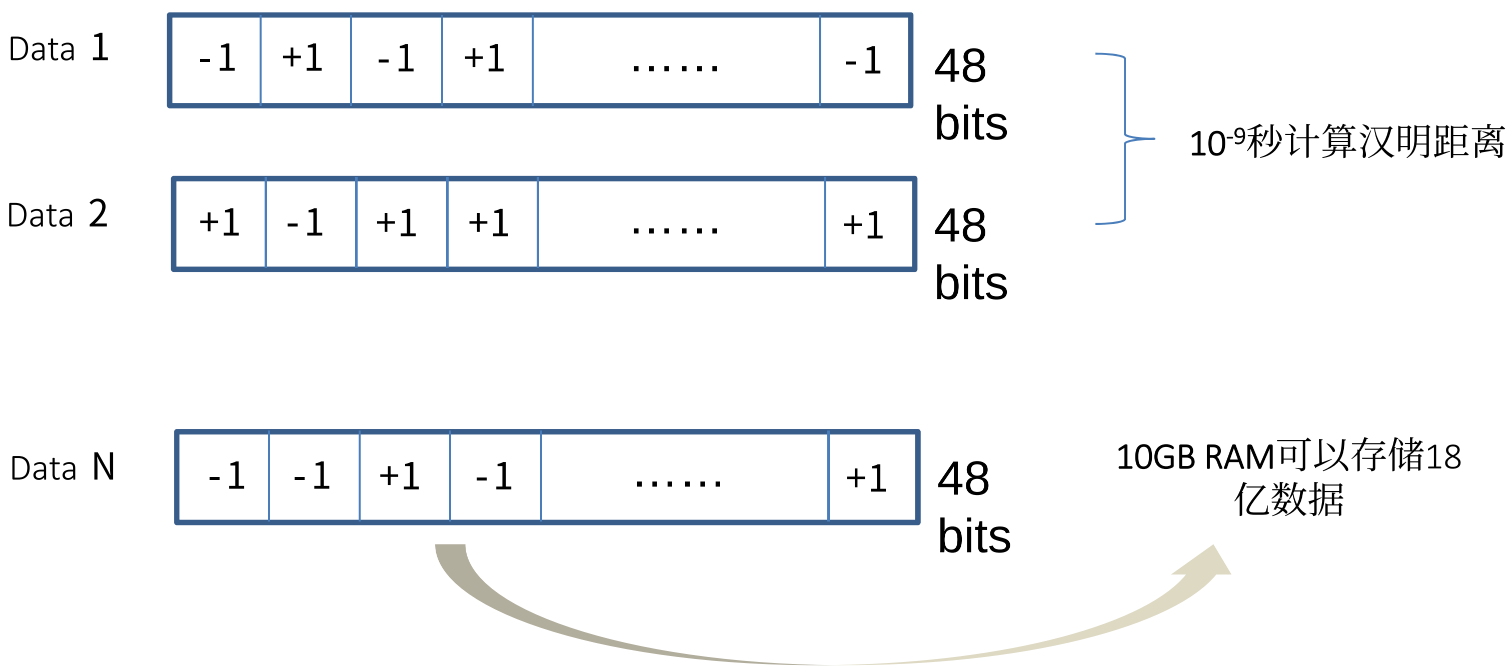
基于哈希的表示

- **What is hashing?**



基于哈希的表示

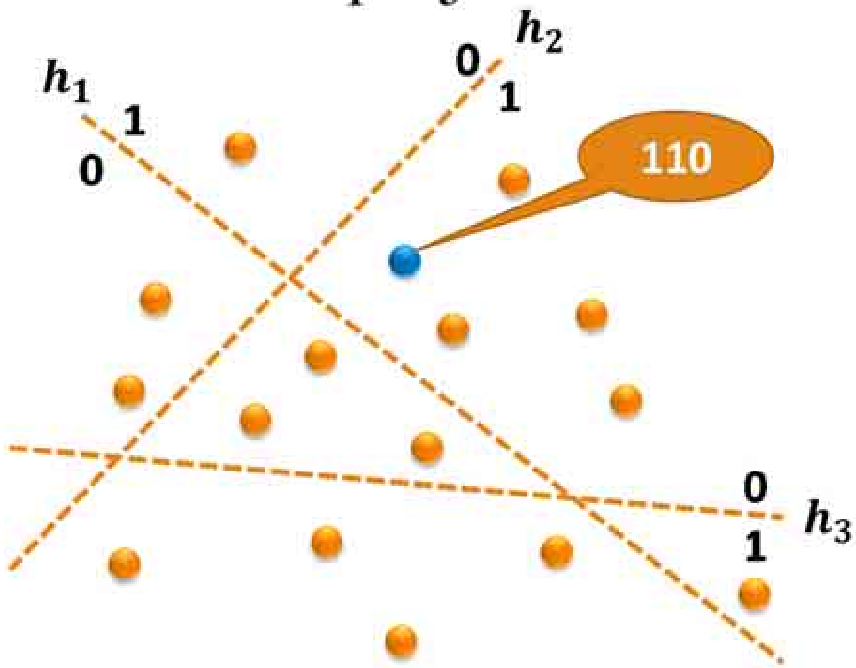
- 时间上高效：基于XOR操作的快速计算
- 存储上高效：基于位存储的紧致表达



基于哈希的表示

- Locality Sensitive Hashing (LSH)
  - ✓ Date-independent, unsupervised
  - ✓ Map the data points with random projections

Hash Function: $B = sgn(XW)$



P.Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. STOC'98

LSH与神经科学

REPORT

A neural algorithm for a fundamental computing problem

Sanjoy Dasgupta[1], Charles F. Stevens[2,3], Saket Navlakha[4,*]
+ See all authors and affiliations

Science 10 Nov 2017:
Vol. 358, Issue 6364, pp. 793-796
DOI: 10.1126/science.aam9868

研究发现果蝇的嗅觉环路对相近的嗅觉产生相近的神经活动模式，可将一种味觉习得的行为应用于接触相似味觉时。研究者将其中的三种全新计算策略应用于计算领域，提出局部敏感哈希算法，该方法可有效改善近似检索的计算表现

LSH与神经科学

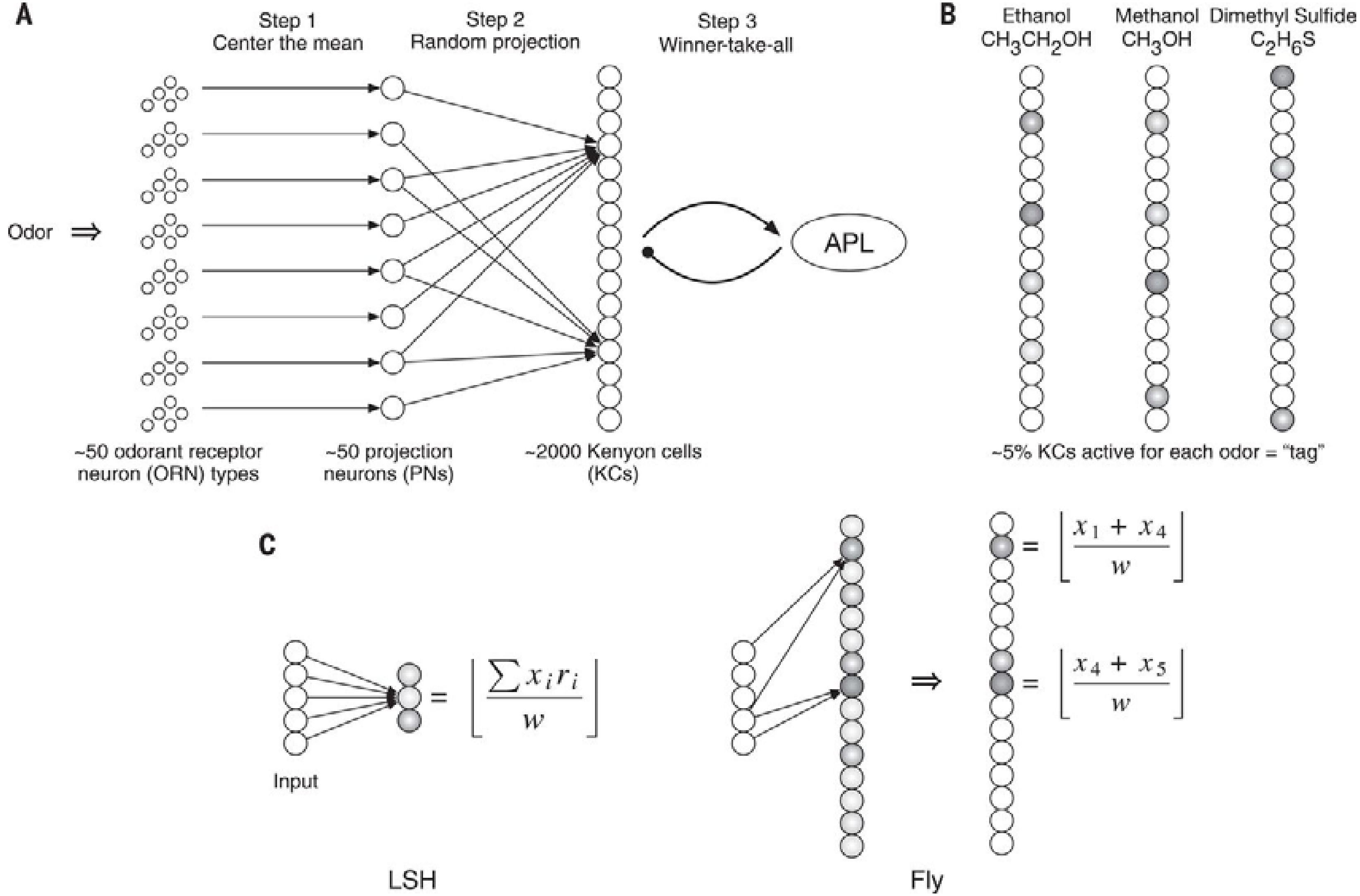


Fig. 1 Mapping between the fly olfactory circuit and locality-sensitive hashing (LSH).

LSH与神经科学

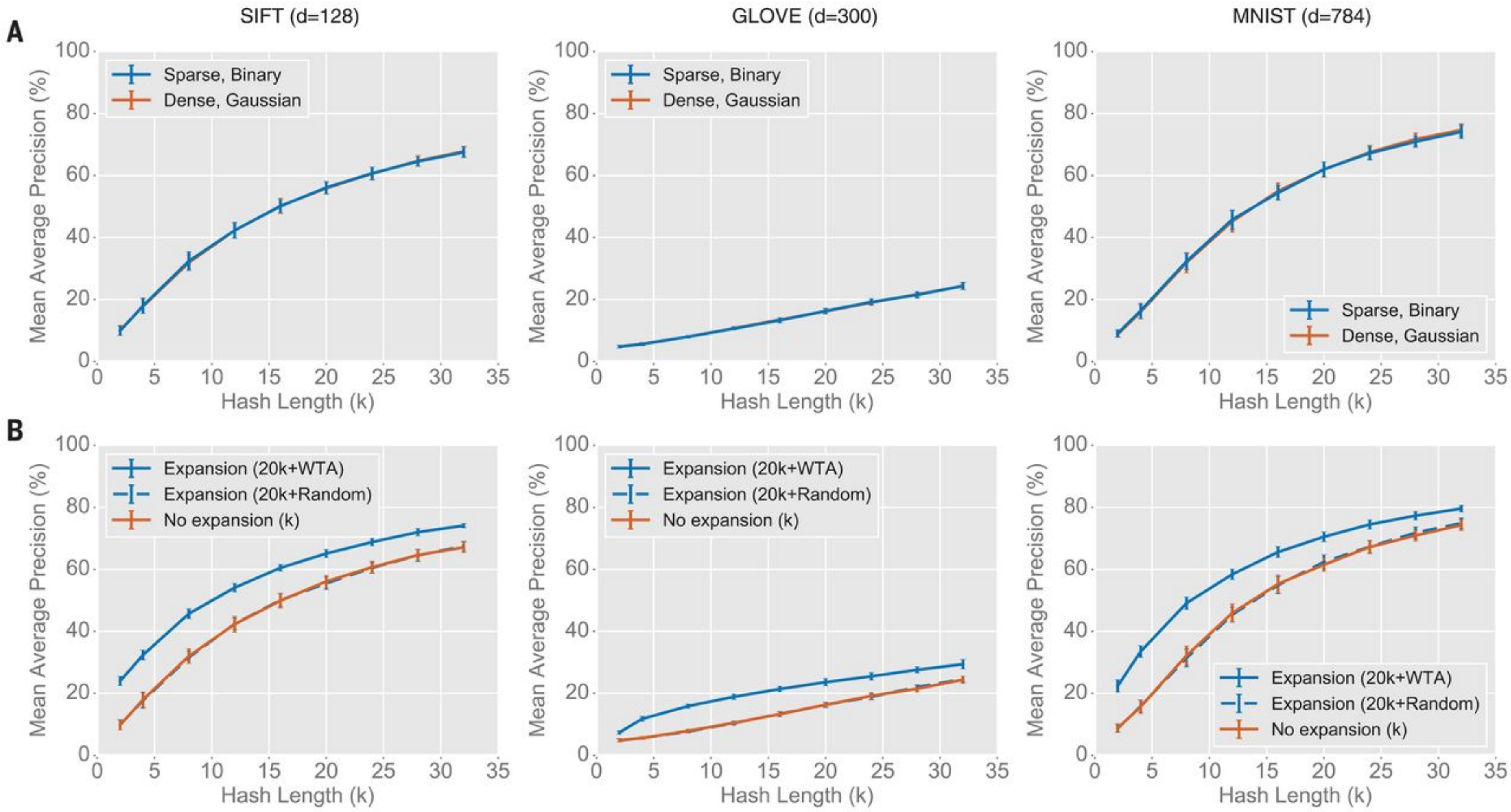


Fig. 2 Empirical comparison of different random projection types and tag-selection methods.

LSH与神经科学

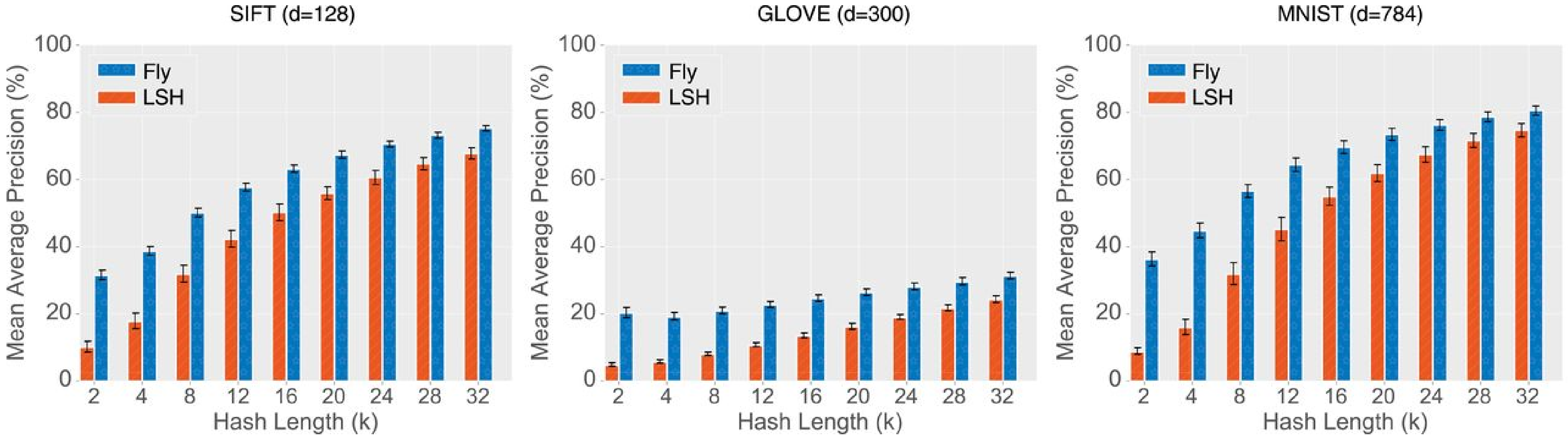
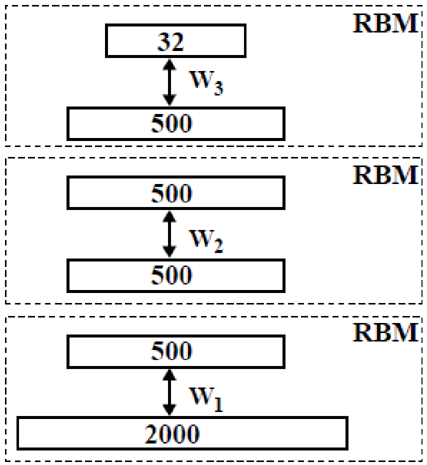


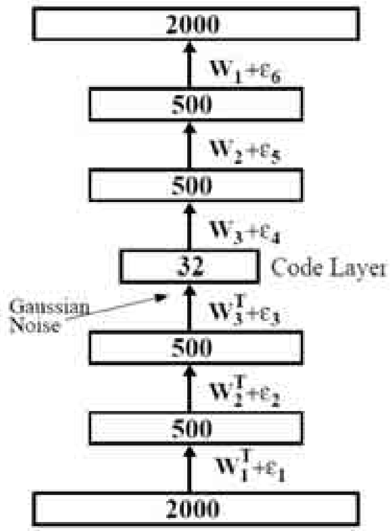
Fig. 3 Overall comparison between the fly algorithm and LSH.

基于哈希的表示

- Restricted Boltzmann Machines (RBMs)
  - ✓ Date-dependent, unsupervised/supervised
  - ✓ Learn the binary codes layer by layer with deep networks

Output layer: final binary codes





R.Salakhutdinov and G.Hinton. Semantic hashing. SIGIR 2007 workshop

基于哈希的表示

- Spectral Hashing (SH)
  - ✓ Date-dependent, unsupervised
  - ✓ Design compact binary codes based on spectral graph partitioning

$$
\begin{aligned}
minimize: &\sum_{ij} W_{ij}\|y_i - y_j\|^2 \\
subject\ to: &\ y_i \in \{-1,1\}^k \\
&\sum_i y_i = 0 \\
&\frac{1}{n}\sum_i y_i y_i^T = I
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
minimize: &\ trace(Y^T(D-W)Y) \\
subject\ to: &\ Y(i,j) \in \{-1,1\} \\
&Y^T 1 = 0 \\
&Y^T Y = I
\end{aligned}
$$

Y. Weiss, A.B. Torralba, and R. Fergus. Spectral hashing. NIPS 2008

基于哈希的表示

- Semi-Supervised Hashing (SSH)
  - ✓ Date-dependent, semi-supervised
  - ✓ Combines the empirical loss over the labeled data with other desirable constraints over both labeled and unlabeled data.

$$\max \ J(W) = \mathrm{tr}\,(W^T X_l S X_l^T W) + \eta * \mathrm{tr}\,(W^T X X^T W)$$

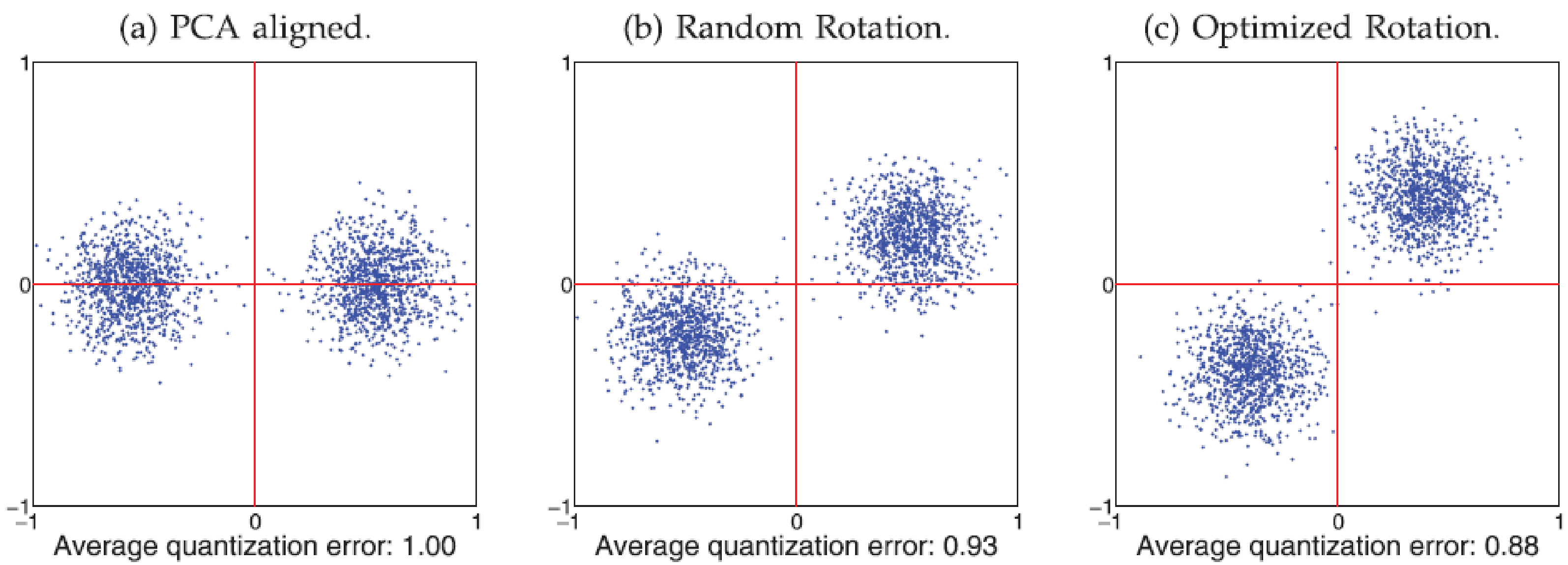
  - ✓ A sequential learning scheme (SPLH) is also developed for hash function leaning.

J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large scale search. TPAMI 2012.

基于哈希的表示

- 迭代量化（Iterative Quantization, ITQ）

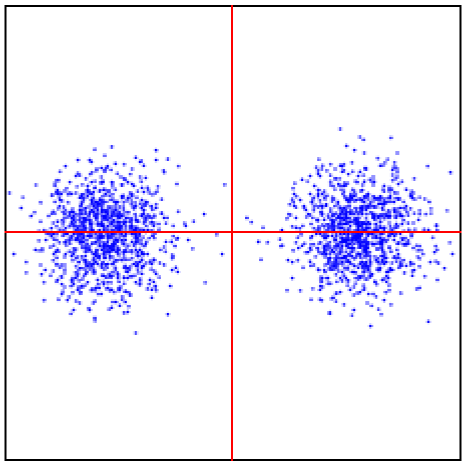
$$\min\|B - XWR\|_F^2, \qquad s.t. R^T R = I$$



Yunchao Gong, et.al.Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-scale Large Retreival. CVPR 2011, IEEE TPAMI 2013.

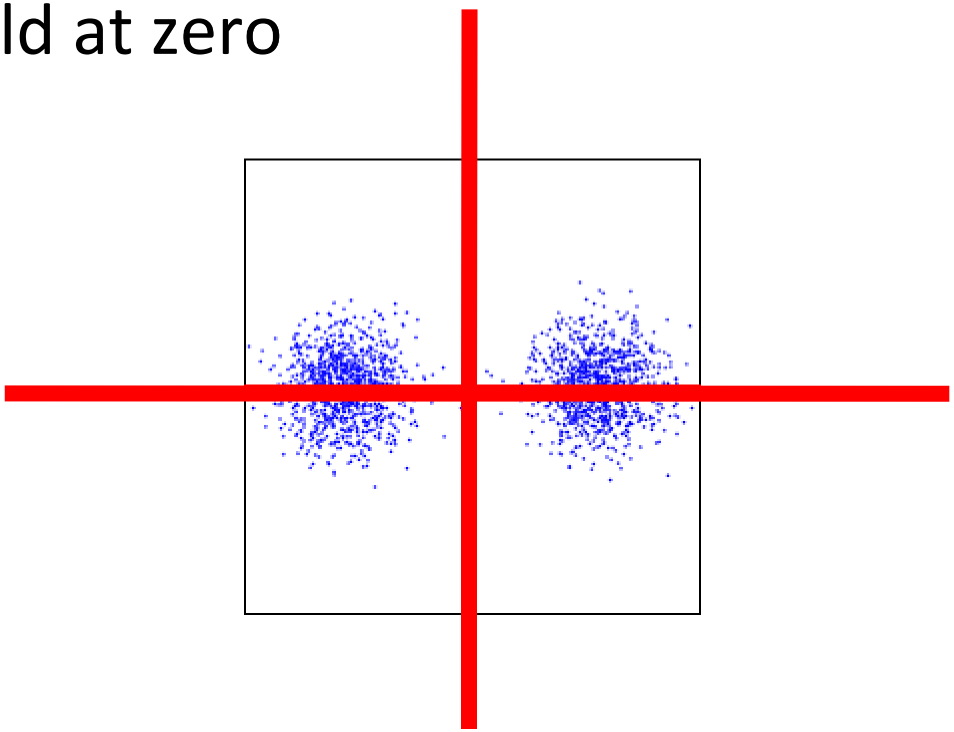
Iterative Quantization(ITQ)

- Baseline scheme:
  - Find PCA embedding of the data
  - For a c-bit code, take top c PCA directions and threshold at zero



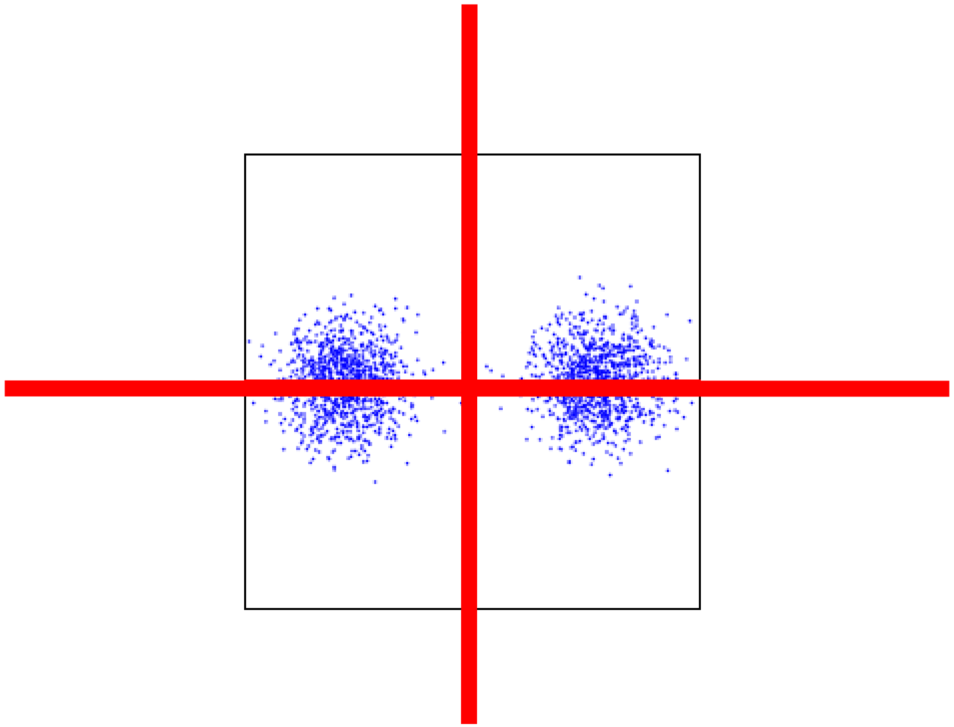
Iterative Quantization(ITQ)

- Baseline scheme:
  - Find PCA embedding of the data
  - For a c-bit code, take top c PCA directions and threshold at zero



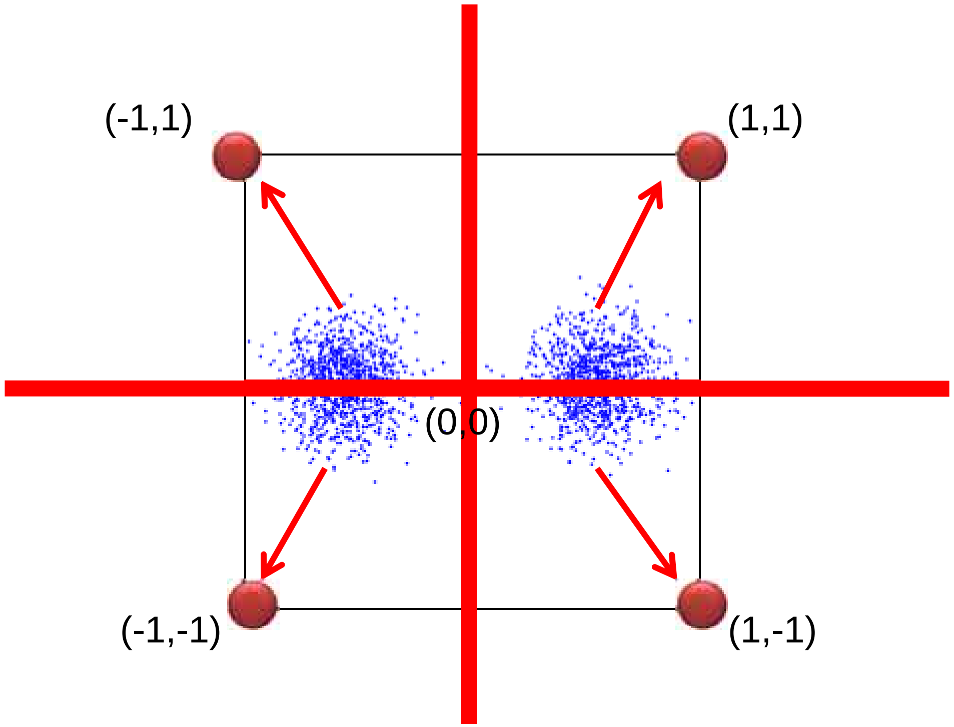
Problem with PCA

- Variance of different dimensions is not balanced



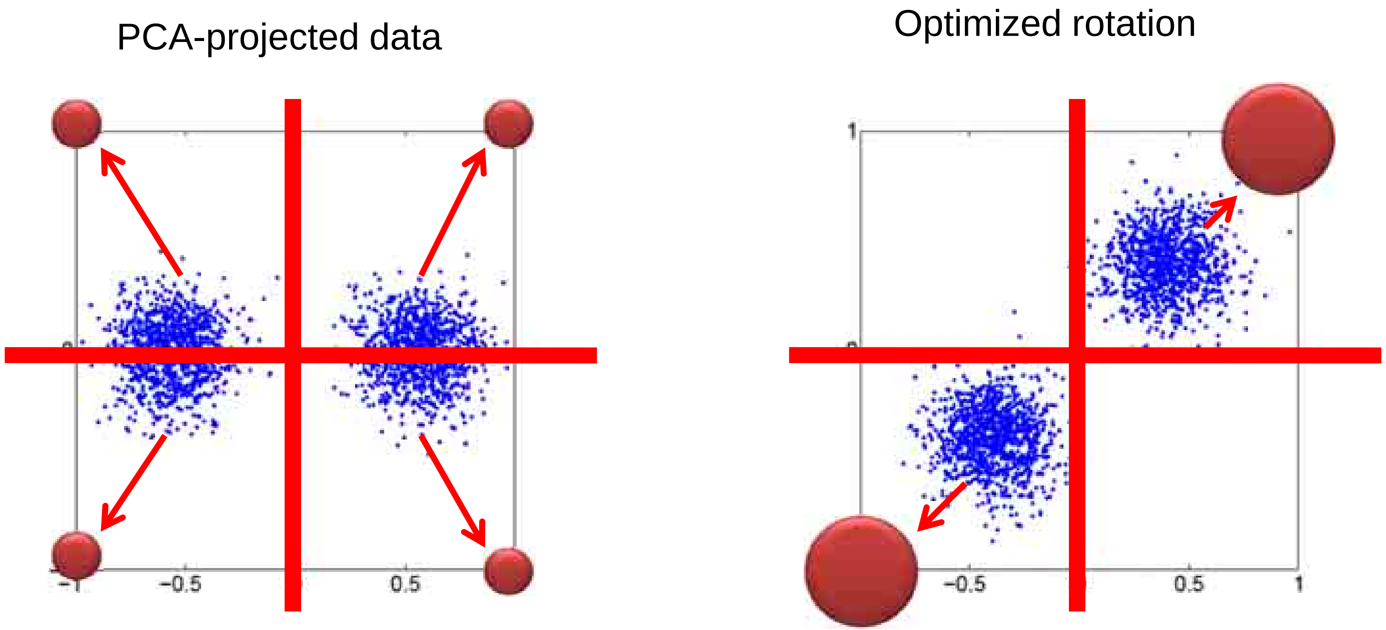
Binary coding as quantization

- **Key idea:** similarity-preserving codes should have low quantization error



Iterative Quantization(ITQ)

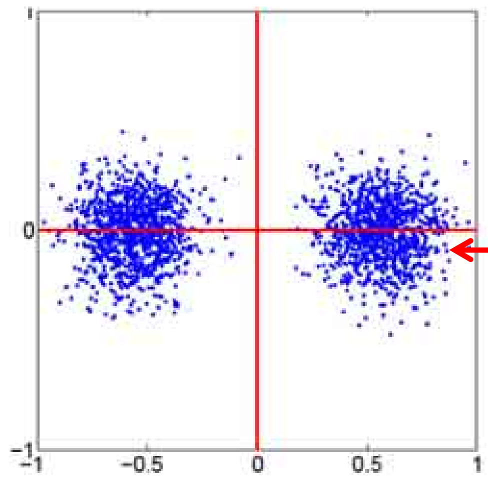
- Rotate the PCA-projected data to minimize quantization error



Iterative Quantization(ITQ)

- Minimize quantization error of PCA-projected data with respect to the Frobenius norm:

$$\min \mathcal{J}(B, R) = \|B - VR\|_F^2$$



PCA-projected data

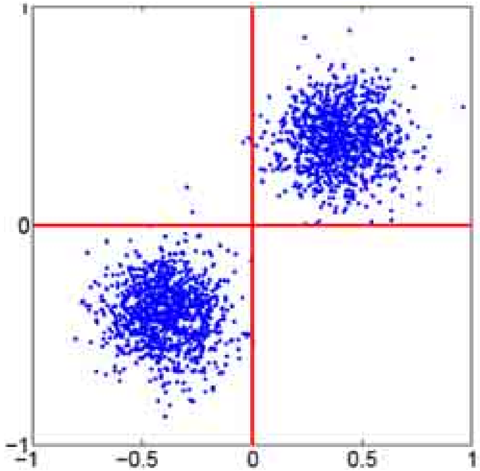
Iterative Quantization(ITQ)

- Minimize quantization error of PCA-projected data with respect to the Frobenius norm:

$$\min \mathcal{J}(B, R) = \|B - VR\|_F^2$$

binary encoding matrix

rotation



Iterative Quantization(ITQ)

- Minimize quantization error of PCA-projected data with respect to the Frobenius norm:

$$\min \mathcal{J}(B,R) = \|B - VR\|_F^2$$
$$subject\ to \quad B \in \{-1,1\}^{n\times c}, \quad R^T R = I$$

  – *B* is an *n* x *c* matrix where each row is the binary string encoding a data point
  – *V* is a matrix of PCA-projected data
  – *R* is a *c* x *c* rotation matrix

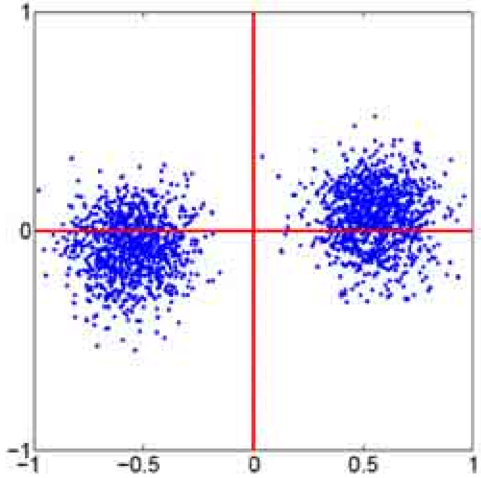
Iterative Quantization(ITQ)

- Minimize quantization error of PCA-projected data with respect to the Frobenius norm:

$$\min \mathcal{J}(B,R) = \|B - VR\|_F^2$$
$$subject\ to \quad B \in \{-1,1\}^{n\times c}, \quad R^T R = I$$

- Alternating minimization:
  - Initialize *R* to a random rotation
  - Fix *R*, solve for *B*
  - Fix *B*, solve for *R*
  - Iterate until convergence

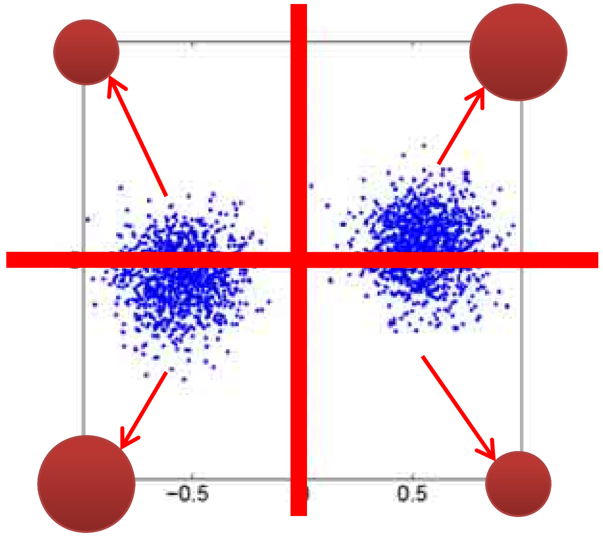
Initialization



randomly rotated data

$$\min \mathcal{J}(R) = \|B - VR\|_F^2,$$

Fix *R*, find *B*



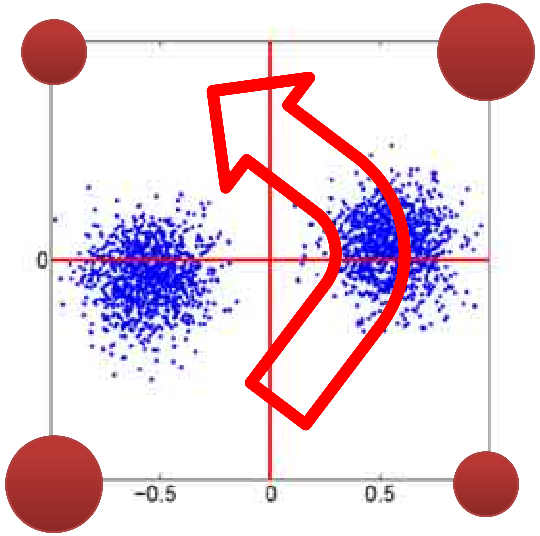
known

$$\min \mathcal{J}(R) = \|B - VR\|_F^2,$$

- Optimal *B* is given by thresholding rotated coordinates:

$$T = VR \qquad \mathcal{J}(B) = \sum_{i=1}^{n}\sum_{j=1}^{c} B_{i,j}T_{i,j}. \quad B_{ij} = \begin{cases} 1, & \text{if } T_{i,j} \geq 0; \\ -1, & otherwise. \end{cases}$$

Fix *B*, find *R*



known

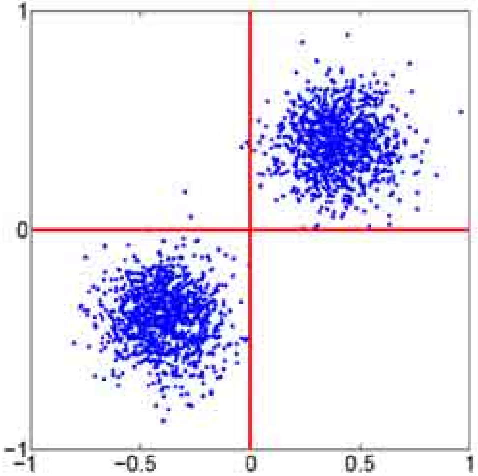
$$\min \mathcal{J}(R) = \|B - VR\|_F^2,$$

- Optimal *R* is found by solving the orthogonal Procrustes problem:

$$\boldsymbol{B}^T\boldsymbol{V} = \boldsymbol{S}\boldsymbol{\Omega}\hat{\boldsymbol{S}}^T \qquad\qquad R = \hat{S}S^T.$$

P. Schonemann. A generalized solution of the orthogonal Procrustes problem. Psychometrika, 31, 1966.

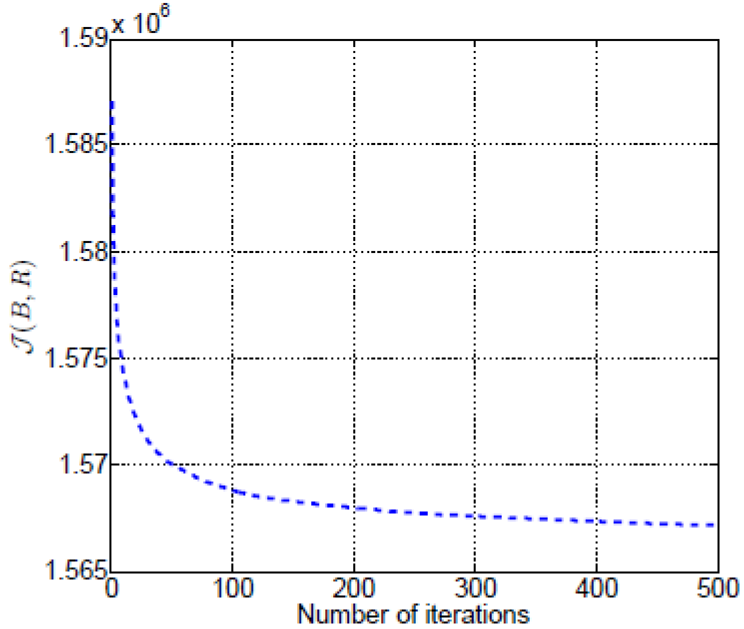
Iterate until convergence



Locally optimal rotation

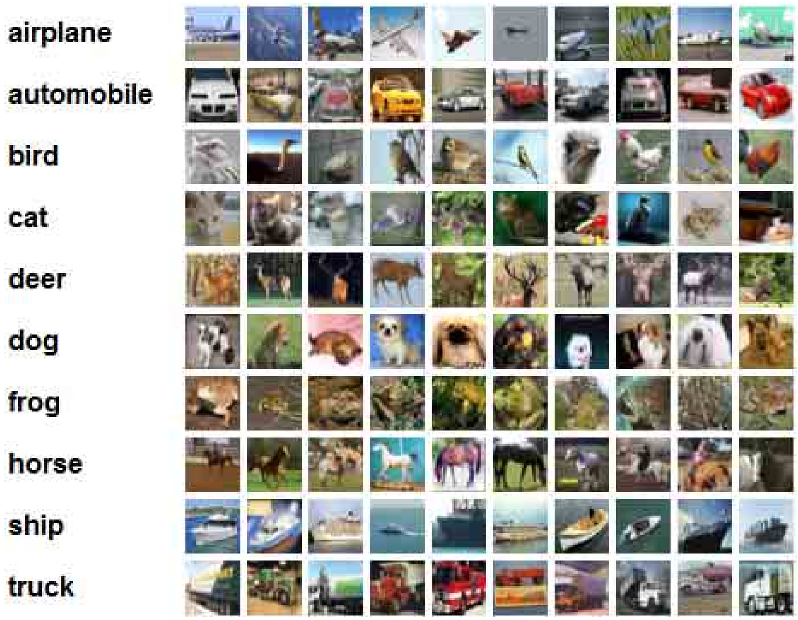
Iterate until convergence

- Behavior of the objective function:



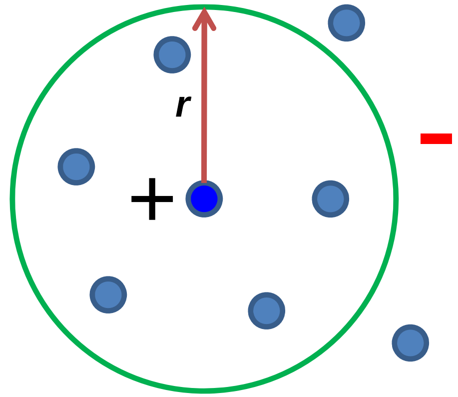
Evaluation

- CIFAR dataset: ~60,000 images, 11 categories
  - "Tiny images" converted to 320-dimensional gist feature vectors

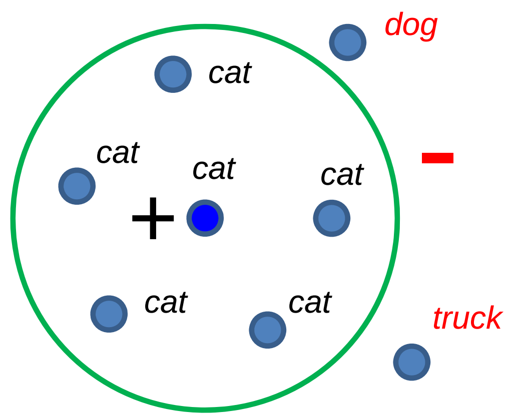


Evaluation

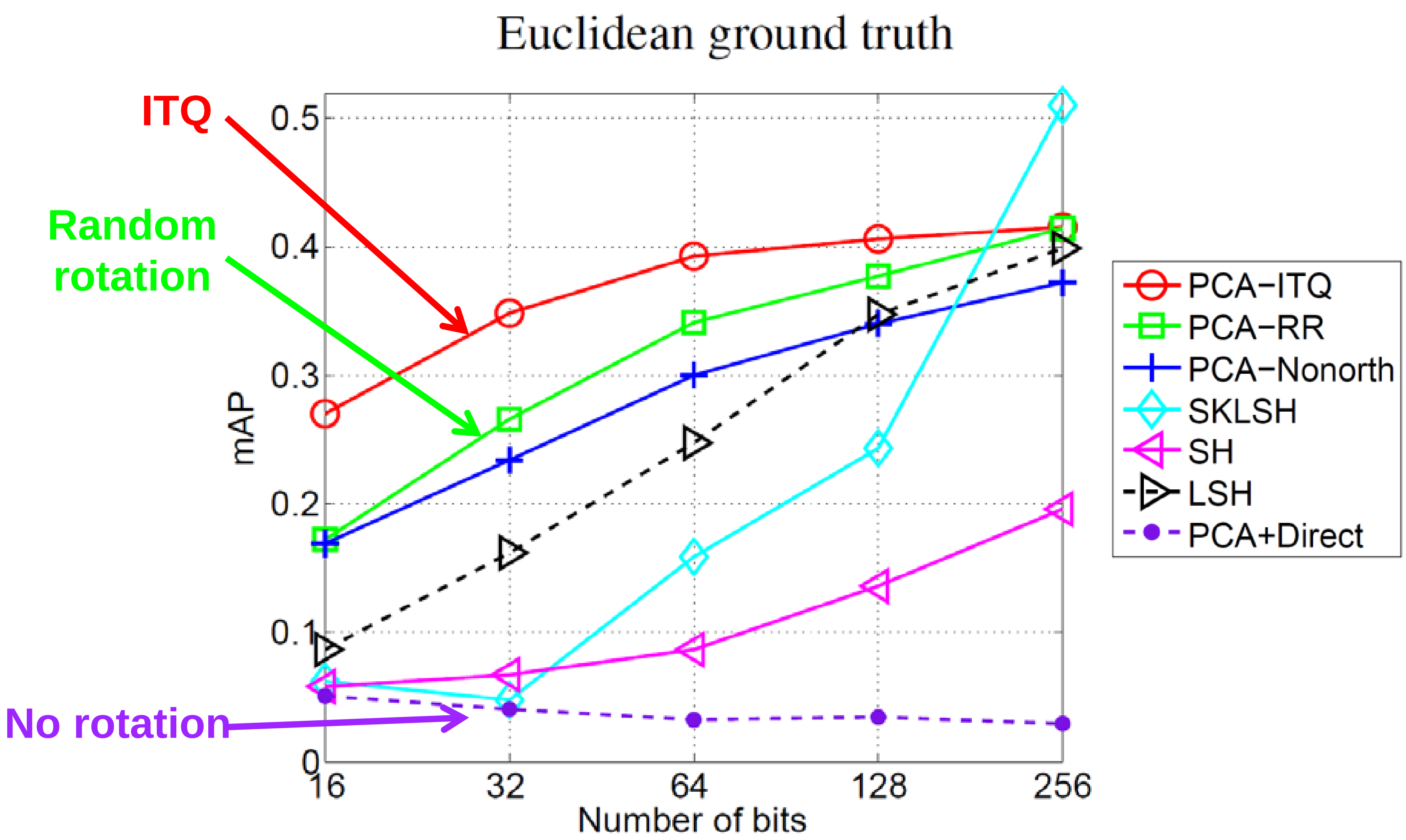
- Euclidean neighbor retrieval
  - Ground truth neighborhood radius defined by average distance to 50th nearest neighbor
  - Performance measured by area under the recall-precision curve (mAP)



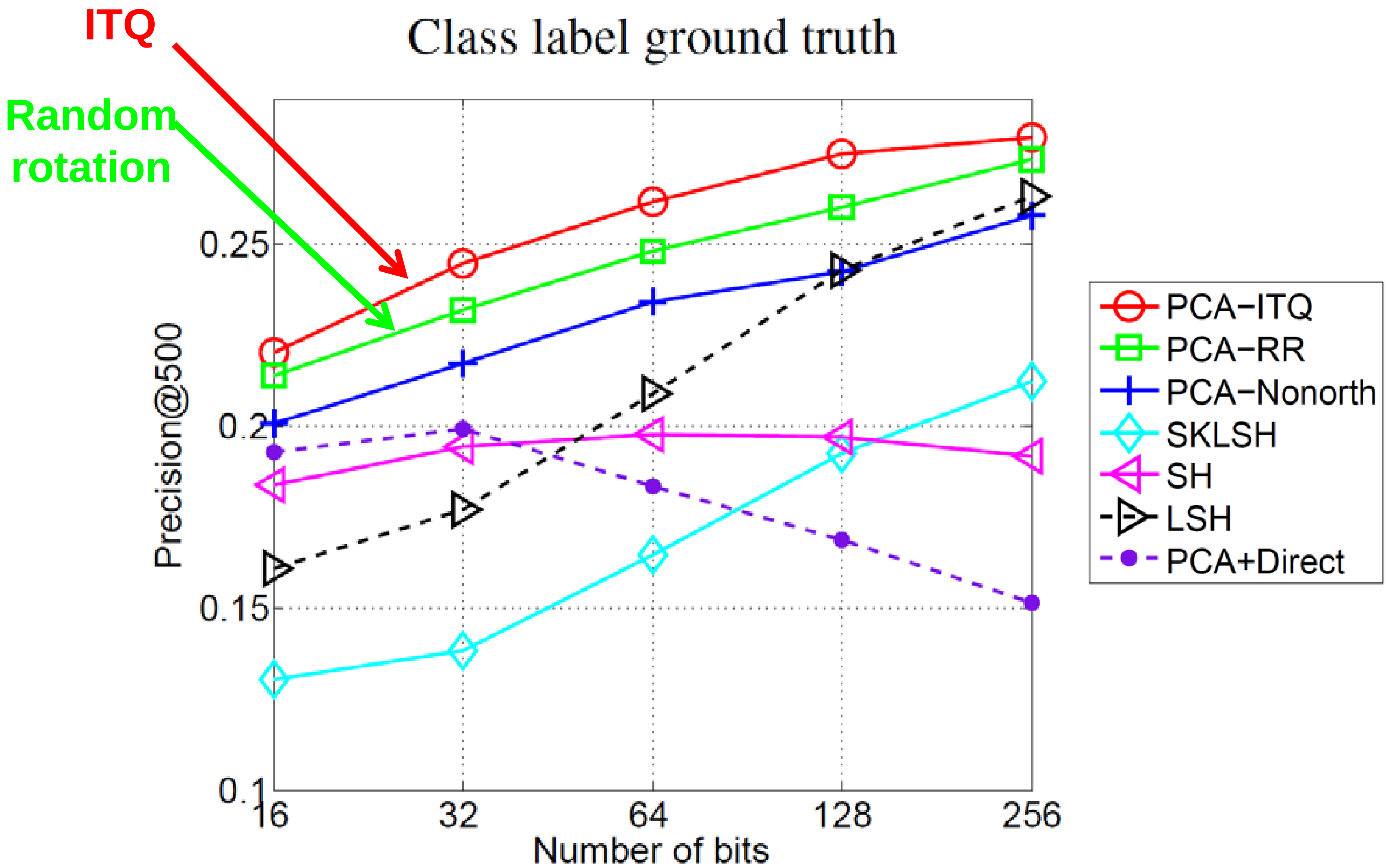
- Semantic neighbor retrieval
  - Ground truth defined by class label
  - Performance measured by average precision of top 500 retrieved matches



Evaluation


Euclidean ground truth
ITQ
Random rotation
No rotation
mAP
0.5
0.4
0.3
0.2
0.1
0
16
32
64
128
256
Number of bits
PCA−ITQ
PCA−RR
PCA−Nonorth
SKLSH
SH
LSH
PCA+Direct

Evaluation


ITQ
Random rotation
Class label ground truth
Precision@500
0.1
0.15
0.2
0.25
16
32
64
128
256
Number of bits
PCA−ITQ
PCA−RR
PCA−Nonorth
SKLSH
SH
LSH
PCA+Direct

在线学习哈希方法

在实际检索系统中数据是流式的、不断更新的

在线更新

大规模

对于超大规模数据，很难将其载入内存进行哈希函数学习

在线学习哈希方法

全部数据 = 之前数据 + 新数据

- 综合所有新老数据，重新学习
- 超高的时间和计算代价

在线学习: Passive-Aggressive (PA) 算法 (JMLR'06)

$$W^t = \arg\min_{W} \frac{1}{2}\|W - W^{t-1}\|_F^2 + C\xi$$

$$\text{s.t. } L(W; \langle(x_i^t, x_j^t), s_{ij}\rangle) \leq \xi \text{ and } \xi \geq 0$$

Online Hashing (IJCAI'13)

Long-Kai Huang, Qiang Yang, Wei-shi Zheng. Online Hashing. IJCAI 2013

在线学习哈希方法

- 矩阵素描(sketching or coreset)

  ✓一个比原始矩阵小的多的矩阵

  ✓保留原始矩阵的大多数特性

$$\forall x,\ \|x\| = 1 \qquad \big|\|P^T x\|^2 - \|Q^T x\|^2\big| \le \varepsilon \|P\|_F^2$$

在线学习哈希方法

Frequent Directions (FD)

Algorithm 1 Frequent Directions (Liberty [21])

Input: Data matrix $P \in \mathbb{R}^{d\times n}$, sketch matrix $Q \in \mathbb{R}^{d\times l}$.
Output: Sketch matrix Q.

```
if Q not exists then
    Q ⟵ all zeros d × l matrix
end if
for each column P_i in P, do
    Insert P_i into a zero valued column of Q
    if Q has no zero valued columns then
        [U, S, V] = SVD(Q)
        \\ C = US    [just for notation]
        Set δ = s²_{l/2}    [the squared (l/2)^th entry of S]
        Set Ŝ = √(max(S² − I_l δ, 0))
        Q = UŜ
    end if
end for
```

Lemma 1. (Liberty [21]) Apply Algorithm 1 to matrix P to obtain a sketch Q with prescribed l, then

$$\forall x,\ \|x\| = 1 \qquad 0 \le \|P^T x\|^2 - \|Q^T x\|^2 \le \frac{2}{l}\|P\|_F^2$$

or

$$0 \le \|PP^T - QQ^T\|_2 \le \frac{2}{l}\|P\|_F^2$$

Edo Liberty, "Simple and Deterministic Matrix Sketching". SIGKDD 2013 (Best paper award)

在线学习哈希方法

广泛意义上的PCA Hashing:

$$\max_{W \in \mathbb{R}^{d \times r}} \quad tr(W^T(X-\mu)(X-\mu)^T W)$$

$$\text{s.t.} \quad W^T W = I_r$$

其中 μ 是全体数据的均值向量.

能否为矩阵 $X-\mu$ 在线维护一个素描矩阵 Y，以至于

$$YY^T \approx (X-\mu)(X-\mu)^T$$

在线学习哈希方法

均值漂移问题:

- 在流式问题中，数据在不断更新, 因此数据的均值 μ 也会不断变化.

给每个数据块加一个虚拟的样本：

- 对于流式数据 $X_t = [D_1, D_2, \ldots, D_t]$, 重新设计数据矩阵 E_t:

$$
\begin{aligned}
E_t = [\mathcal{D}_1 - \overline{\mathcal{D}_1}, \quad & \mathcal{D}_2 - \overline{\mathcal{D}_2}, \sqrt{\frac{n_1 m_2}{n_1 + m_2}}(\overline{\mathcal{D}_2} - \mu_1), \cdots, \\
& \mathcal{D}_i - \overline{\mathcal{D}_i}, \sqrt{\frac{n_{i-1} m_i}{n_{i-1} + m_i}}(\overline{\mathcal{D}_i} - \mu_{i-1}), \cdots, \\
& \mathcal{D}_t - \overline{\mathcal{D}_t}, \sqrt{\frac{n_{t-1} m_t}{n_{t-1} + m_t}}(\overline{\mathcal{D}_t} - \mu_{t-1})]
\end{aligned}
$$

在线学习哈希方法

- 基于以上的设计，在任意适合 t, 可以证明：

$$E_t E_t^T = cov(X_t)$$

- 通过为新设计的矩阵 E_t 在线维护素描矩阵 Y, 我们可以基于这个新的小矩阵 Y学习哈希函数.

Algorithm 2 Zero Mean Sketching

Input: Streaming data chunk $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_k$,
All zeros matrix Y of size $d \times l$.

1: Sketch $\mathcal{D}_1 - \overline{\mathcal{D}_1}$ into Y with Algorithm 1
2: $n \leftarrow m_1$ and $\mu \leftarrow \overline{\mathcal{D}_1}$
3: **for** $i = 2 : k$, **do**
4: Sketch $[\mathcal{D}_i - \overline{\mathcal{D}_i}, \sqrt{\frac{nm_i}{n+m_i}}(\overline{\mathcal{D}_i} - \mu)]$ into Y
5: $\mu \leftarrow \frac{n\mu}{n+m_i} + \frac{m_i\overline{\mathcal{D}_i}}{n+m_i}$ [update the data mean]
6: $n \leftarrow n + m_i$ [update the data size]
7: **end for**

对比实验

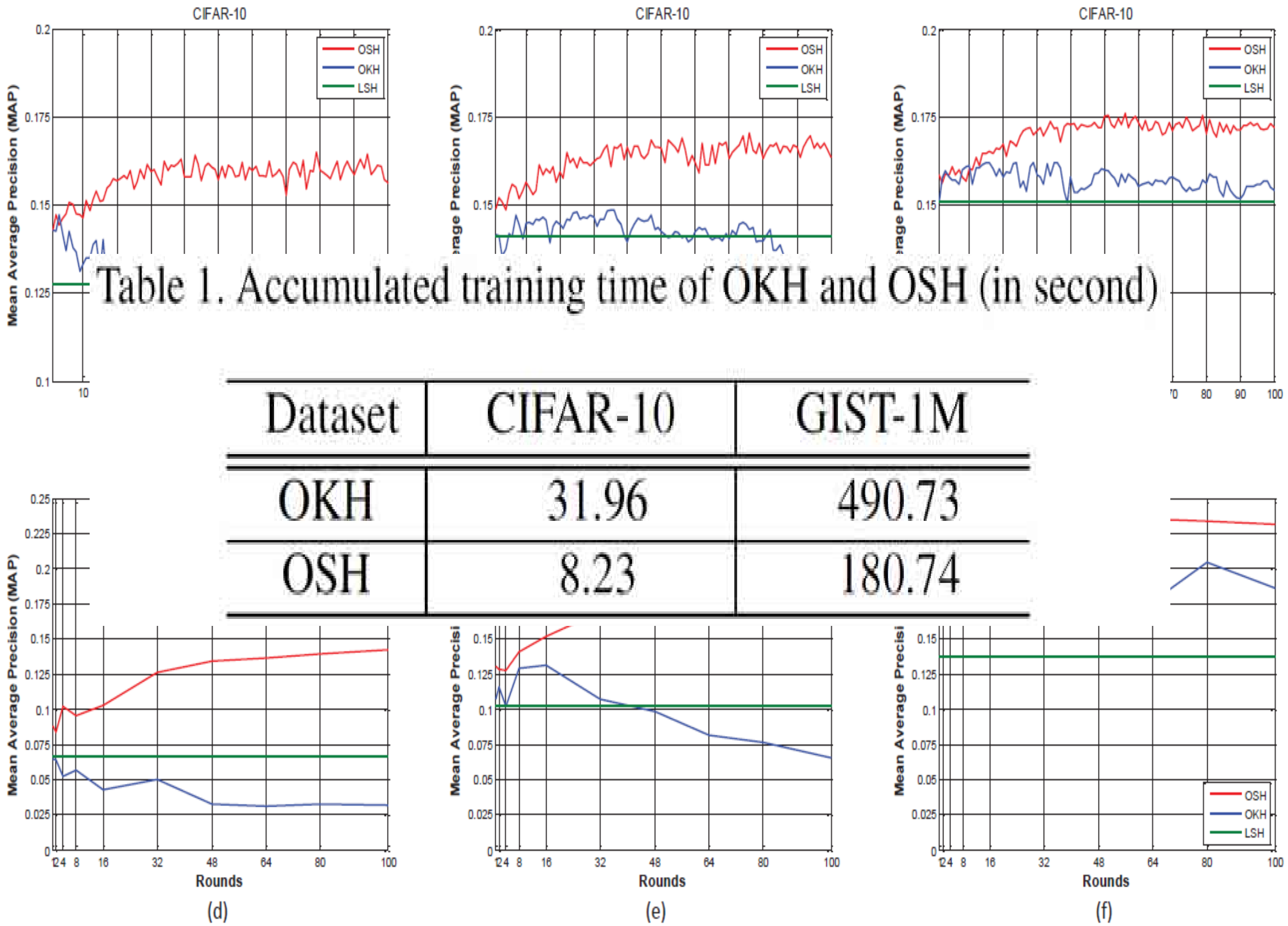
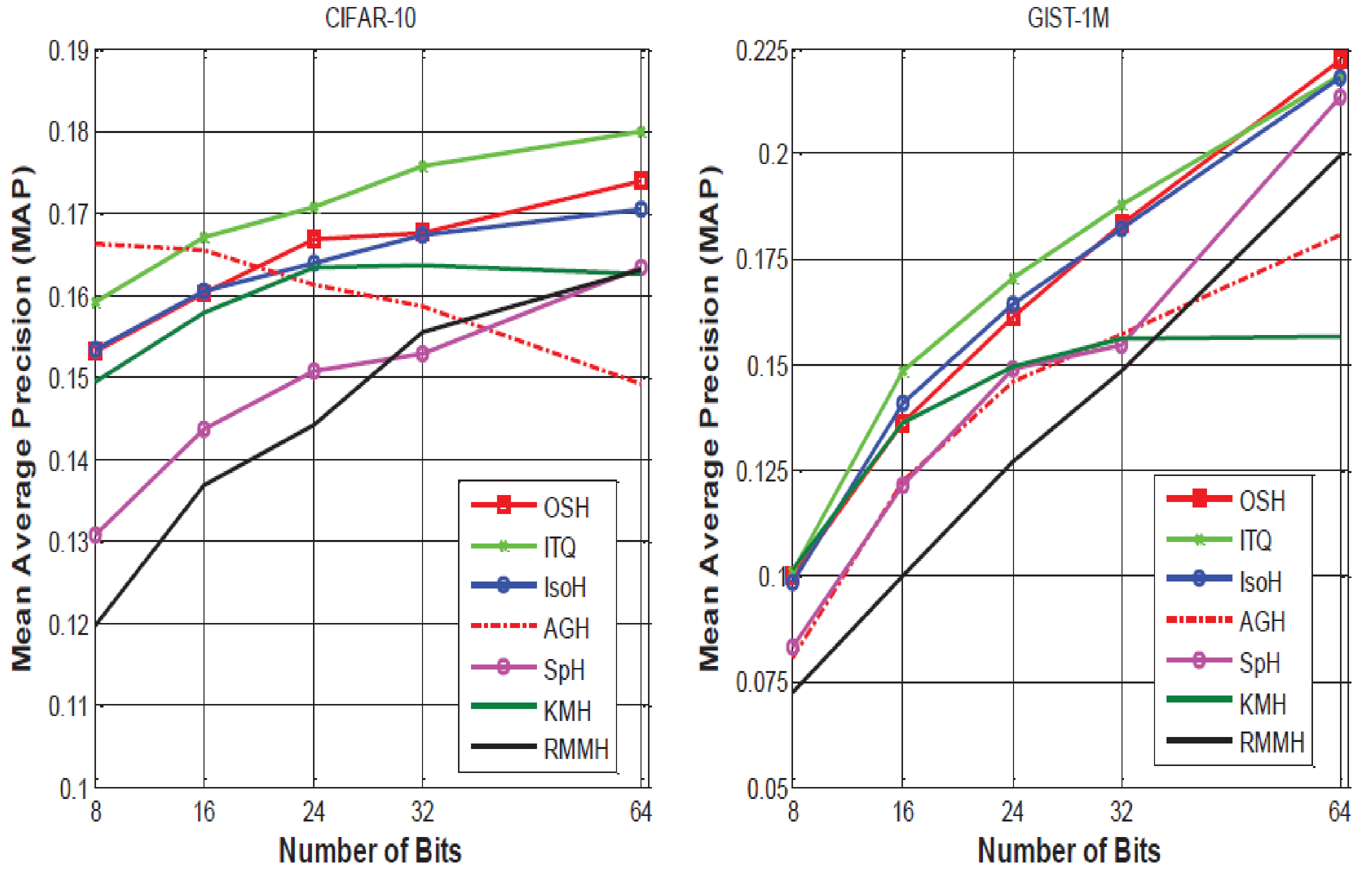


Table 1. Accumulated training time of OKH and OSH (in second)

Dataset	CIFAR-10	GIST-1M
OKH	31.96	490.73
OSH	8.23	180.74

Figure 3. (a)(b)(c) Mean average precision (MAP) on CIFAR-10 dataset at each round with 16, 32, 64 bits. (d)(e)(f) MAP on GIST-1M dataset at each round with 16, 32, 64 bits. (Best viewed in color)

对比实验

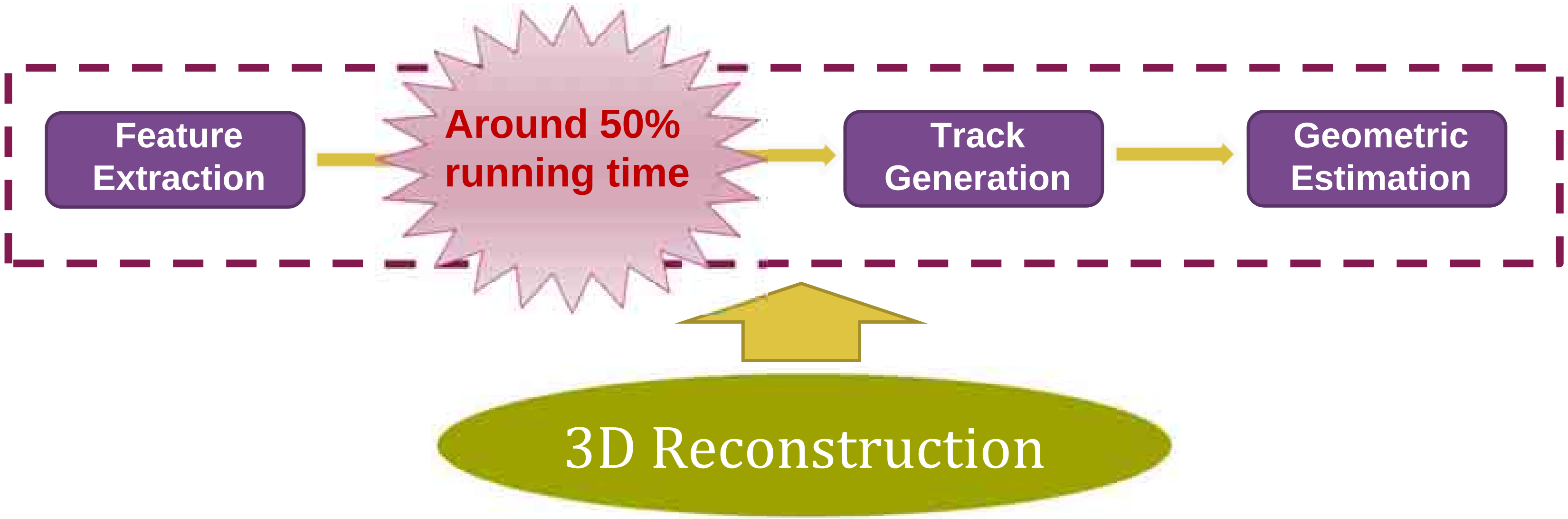


(a) (b)

Cong Leng et al., "Online Sketching Hashing". CVPR 2015

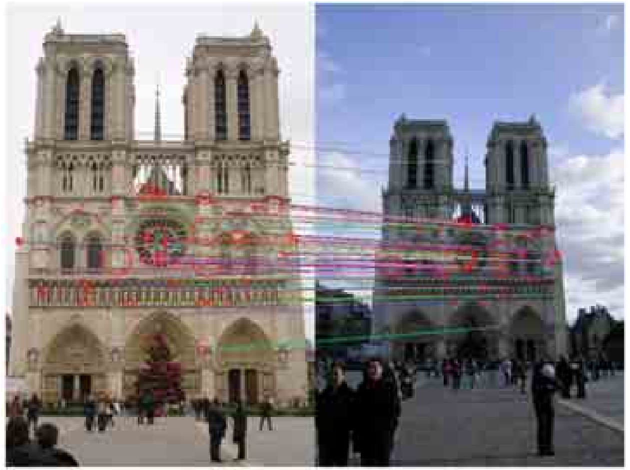
基于哈希的特征匹配

- 三维重建主要包括以下四步:



# Images	# Cores	Match Time	Reconstruction Time	Largest Component
150,000	496	13 Hours	8 Hours	2,106

基于哈希的特征匹配



特征匹配

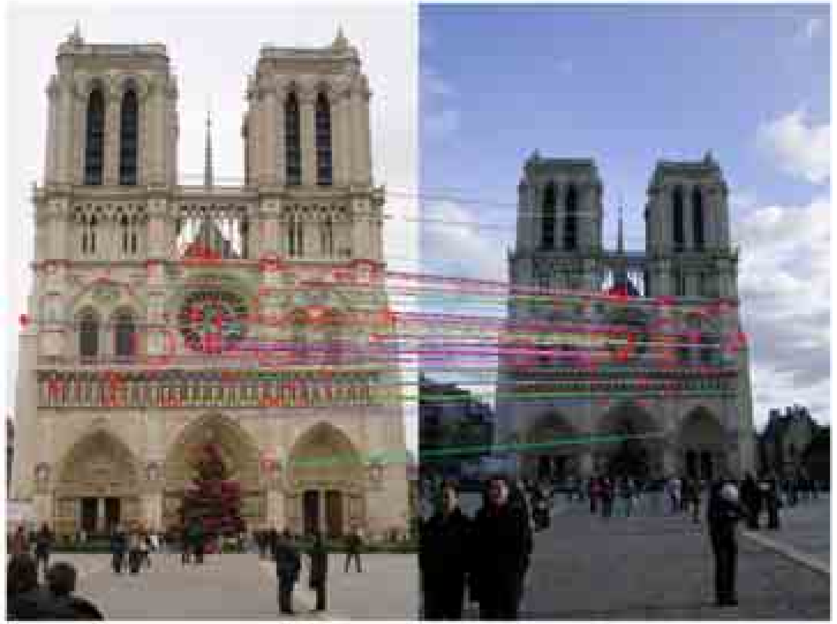
- 图像搜索
- 三维重建
- 图像分类
- 目标识别

典型应用

- **匹配复杂度太高**：$O(N*(N-1)*M^2)$
- **硬件加速成本高**：GPU、并行

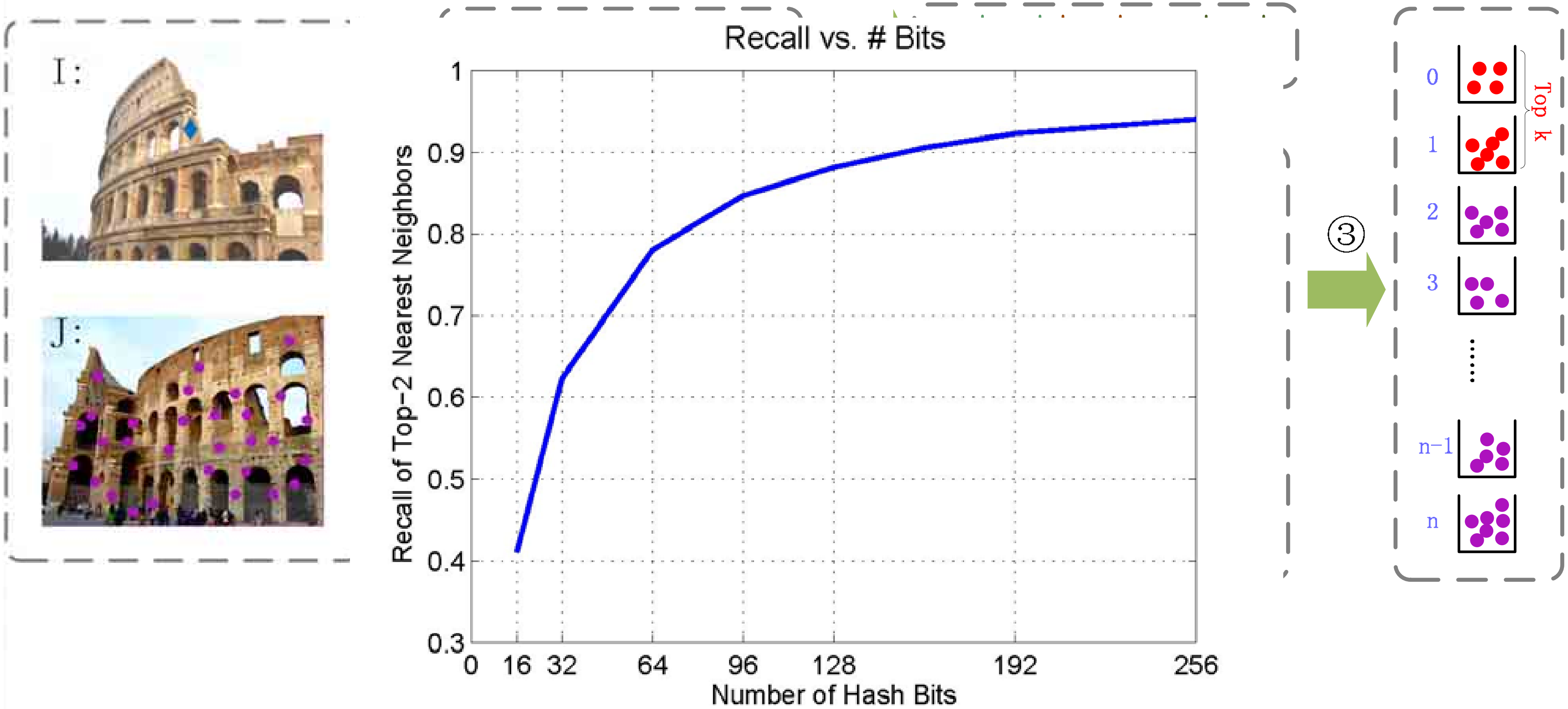
基于哈希的特征匹配

- 现有特征匹配方法可以分为三大类:
  - Point matching
  - Line matching
  - Region matching



Point matching is searching in essence!

基于哈希的特征匹配



Jian Cheng et al., “Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction ”. CVPR 2014

基于哈希的特征匹配

(a) Taj.Mahal: 189 images.

(b) Statue.of.Liberty: 674 images.

(c) Notre.Dame: 715 images.

(d) Colosseum: 1357 images.

基于哈希的特征匹配

Method	Taj.Mahal			Statue.of.Liberty		
	T_{Match}	Speed-Up	Points	T_{Match}	Speed-Up	Points
Brute	52575s	1.00×	**124038**	30608s	1.00×	88001
KD						3674
LDA						3274
CasH						3587
CasHa						**6206**

比brute force方法快上百倍

比通用的k-d tree方法快10-40倍

Me						
	T_{Match}	Speed-Up	Points	T_{Match}	Speed-Up	Points
Brute	396729s	1.00×	358121	12307s	1.00×	**540308**
KDTree	60663s	6.54×	347056	2430s	5.06×	445774
LDAHash	13136s	30.20×	413348	851s	14.46×	492040
CasHash-8Bit	2266s	175.08×	**484960**	222s	55.44×	393408
CasHash-10Bit	**1354s**	**293.01**×	400673	**196s**	**62.79**×	512508

基于哈希的特征匹配

OpenMVG (open Multiple View Geometry)

openMVG

通过测试确信它在不同数据集上都非常有效

I'm actually testing it to ensure it works well on different datasets and that I have a code that works as good as your original version.

https://github.com/openMVG/openMVG/issues/194

Four Eyes Lab
Department of Computer Science
University of California, Santa Barbara

[Theia Vision Library](https://github.com/kip622/Theia)

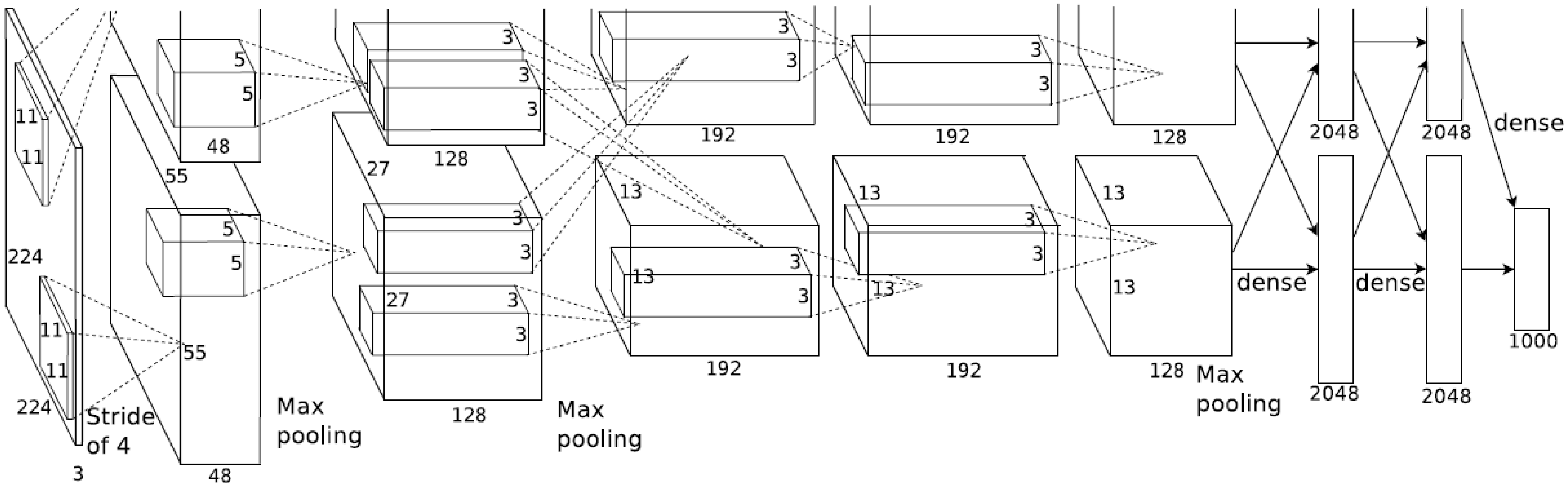
这是非常有用的见解和足够简单的想法

it's a really useful insight and is a simple enough idea.

https://github.com/kip622/Theia

基于量化的CNN加速与压缩

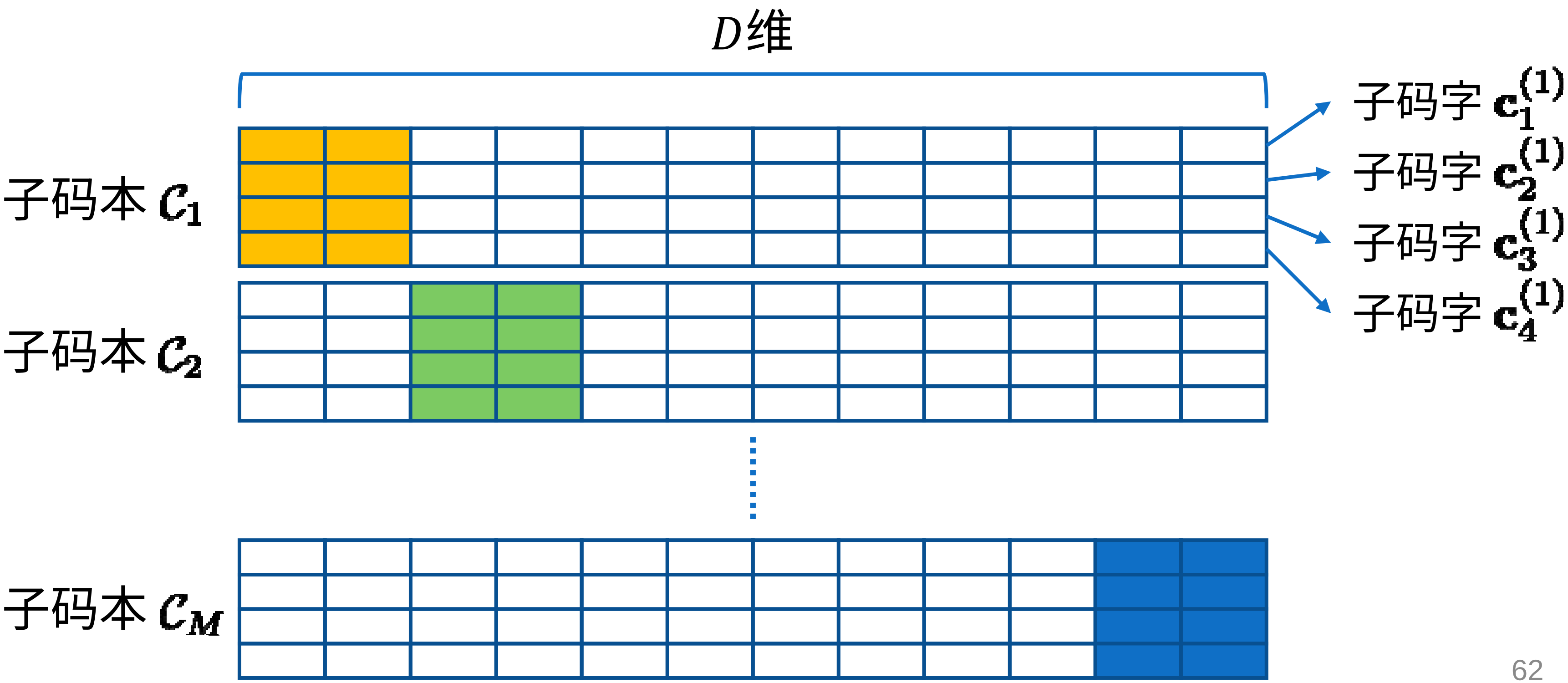
- 卷积神经网络（convolutional neural network）
  – 特征表征能力更强



Hash—>PQ

- 乘积量化（product quantization, PQ）
  - 将样本特征量化为M个子码字之和的形式

$$q(\mathbf{x}) = \sum_{m=1}^{M} q_m(\mathbf{x}),\ q_m(\mathbf{x}) \in \mathcal{C}_m$$



Hash—>PQ

- 乘积量化
  - 优化目标：最小化量化误差

$$\min_{C_1,\ldots,C_M} \sum_{\mathbf{x}} \|\mathbf{x} - q(\mathbf{x})\|_2^2$$

  - 可等价为分别优化各个子空间下的子码本

$$\min_{C_m} \sum_{\mathbf{x}} \|u_m(\mathbf{x}) - q_m(\mathbf{x})\|_2^2$$

  - 求解方法：k均值聚类
  - 时间复杂度：$O(TkND)$

难以应用于大规模数据集

基于量化的CNN加速与压缩

Fully-connected Layer:

$$T(c_t) = \langle W_{c_t}, S \rangle$$

Convoluational Layer:

$$T_{p_t}(c_t) = \sum_{(p_k, p_s)} \langle W_{c_t, p_k}, S_{p_s} \rangle$$

基于量化的CNN加速与压缩

For the fully-connected layer, we split the weighting vector W_{c_t} and layer input S into M sub-vectors, each of $C'_s = C_s/M$ dimensions:

$$T(c_t) = \langle W_{c_t}, S \rangle = \sum_{m=1}^{M} \langle W_{c_t}^{(m)}, S^{(m)} \rangle$$

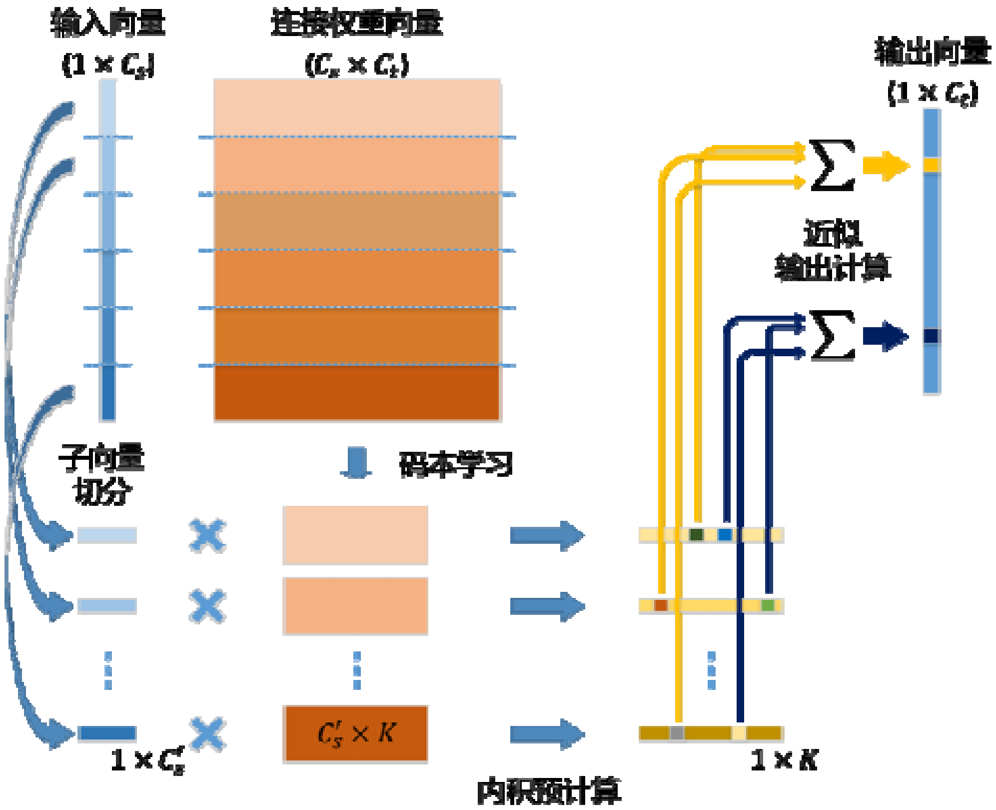
We quantize each sub-vector $W_{c_t}^{(m)}$ with:

$$W_{c_t}^{(m)} \approx D^{(m)} B_{c_t}^{(m)}$$

where $W_{c_t}^{(m)} \in \mathbb{R}^{C'_s \times 1}$, $D^{(m)} \in \mathbb{R}^{C'_s \times K}$, and $B_{c_t}^{(m)} \in \{0, 1\}^{K \times 1}$

基于量化的CNN加速与压缩

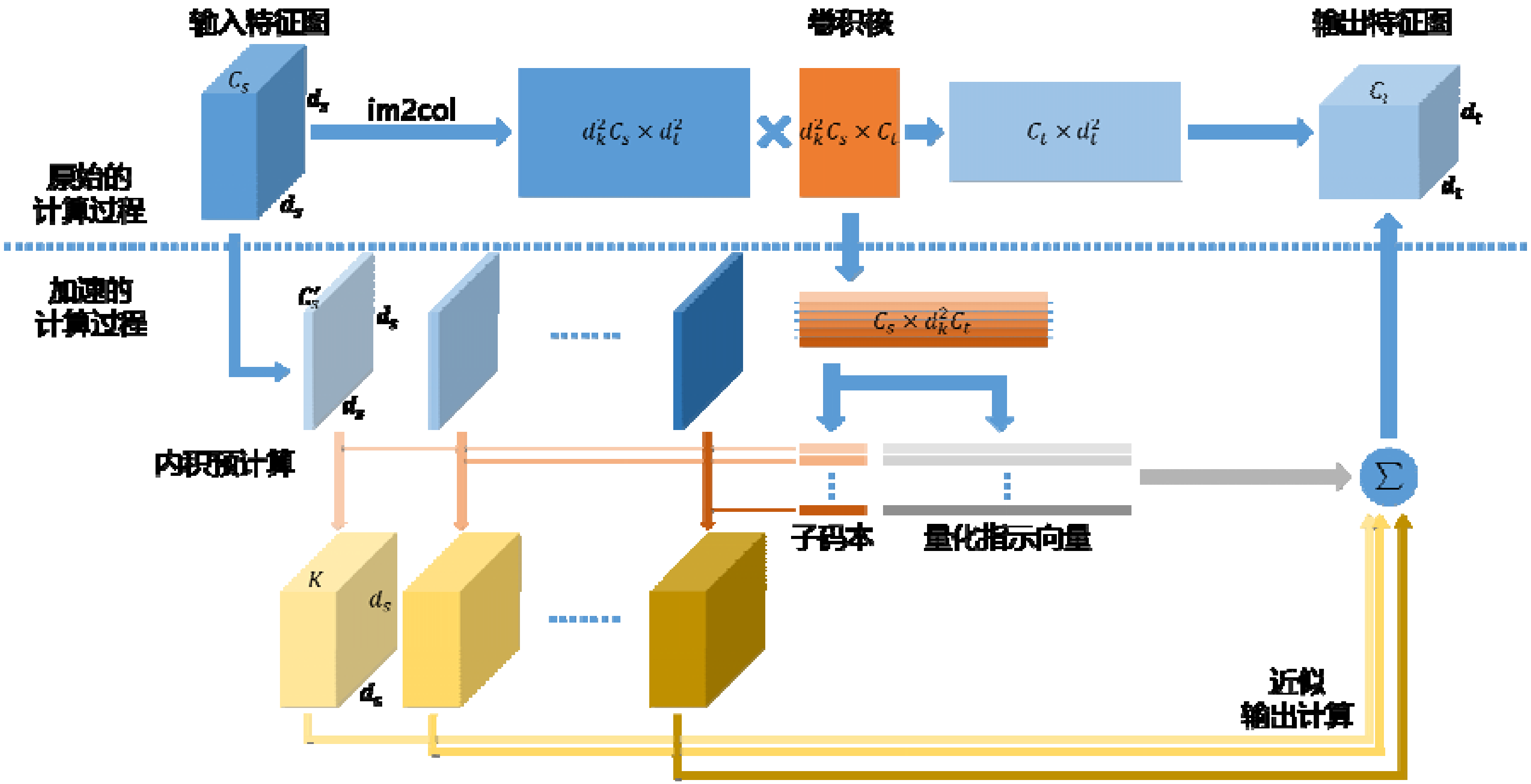
- 全连接层的参数量化与加速计算



Jiaxiang Wu, et al., "Quantized Convolutional Neural Networks for Mobile Devices". CVPR 2016

基于量化的CNN加速与压缩

- 卷积层的参数量化与加速计算



Jiaxiang Wu, et al., “Quantized Convolutional Neural Networks for Mobile Devices”. CVPR 2016

基于量化的CNN加速与压缩

- 参数量化的求解方法
  - 最小化网络参数的量化误差

$$\min_{\mathbf{D},\mathbf{B}} \left\| \mathbf{W} - \tilde{\mathbf{W}} \right\|_F^2$$

$$\text{s.t. } \tilde{\mathbf{W}} = g(\mathbf{D}, \mathbf{B})$$

  - 最小化网络输出的近似误差

$$\min_{\mathbf{D},\mathbf{B}} \left\| f(\mathbf{X}; \mathbf{W}) - f\left(\mathbf{X}; \tilde{\mathbf{W}}\right) \right\|_F^2$$

$$\text{s.t. } \tilde{\mathbf{W}} = g(\mathbf{D}, \mathbf{B})$$

Jiaxiang Wu, et al., "Quantized Convolutional Neural Networks for Mobile Devices". CVPR 2016

基于量化的CNN加速与压缩

- 累积误差问题
  - 网络中浅层的参数被量化后，会改变深层的输入

$$\min_{\mathbf{D},\mathbf{B}} \left\| f(\mathbf{X};\mathbf{W}) - f\left(\mathbf{X};\tilde{\mathbf{W}}\right) \right\|_F^2$$
$$\text{s.t. } \tilde{\mathbf{W}} = g(\mathbf{D},\mathbf{B})$$

累积误差修正

$$\min_{\mathbf{D},\mathbf{B}} \left\| f(\mathbf{X};\mathbf{W}) - f\left(\tilde{\mathbf{X}};\tilde{\mathbf{W}}\right) \right\|_F^2$$
$$\text{s.t. } \tilde{\mathbf{W}} = g(\mathbf{D},\mathbf{B})$$

Jiaxiang Wu, et al., “Quantized Convolutional Neural Networks for Mobile Devices”. CVPR 2016

基于量化的CNN加速与压缩

- 分类性能对比
  - 数据集：MNIST

方法	三层网络		五层网络	
	压缩倍数	分类错误率	压缩倍数	分类错误率
原始网络	-	1.35%	-	1.12%
RER [33]	8.0×	2.19%	8.0×	1.24%
LRD [165]	8.0×	1.89%	8.0×	1.77%
DK [50]	8.0×	1.71%	8.0×	1.26%
NN-ES [48]	8.0×	1.69%	8.0×	1.35%
HashNets [48]	8.0×	1.43%	8.0×	1.22%
Q-CNN[1]	10.9×	1.42%	13.0×	1.34%
Q-CNN[2]	10.9×	1.39%	13.0×	1.19%

Jiaxiang Wu, et al., "Quantized Convolutional Neural Networks for Mobile Devices". CVPR 2016

基于量化的CNN加速与压缩

- 分类性能对比
  - 数据集：ILSVRC-12

网络	方法	超参		加速倍数	压缩倍数	Top-1 分类错误率 ↑	Top-5 分类错误率 ↑
		卷积层	全连接层				
AlexNet	BC [43]	-	-	2.00×	32.00×	21.20%	19.20%
	BWN [46]	-	-	2.00×	32.00×	2.80%	3.20%
	DC [38]	-	-	-	9.00/35.00×	0.00%	-0.03%
	Q-CNN^2	8/128	3/32	4.05×	15.10×	1.38%	0.84%
		8/128	4/32	4.15×	18.32×	1.46%	0.97%
CaffeNet	Q-CNN^2	8/128	3/32	4.04×	15.10×	1.43%	0.99%
		8/128	4/32	4.16×	18.32×	1.54%	1.12%
CNN-S	Q-CNN^2	8/128	3/32	5.69×	15.93×	1.48%	0.81%
		8/128	4/32	5.78×	19.57×	1.64%	0.85%
VGG-16	DC [38]	-	-	-	13.00/49.00×	-0.33%	-0.41%
	Q-CNN^2	8/128	3/32	4.92×	16.06×	1.02%	0.38%
		8/128	4/32	4.94×	19.60×	1.13%	0.45%

Jiaxiang Wu, et al., “Quantized Convolutional Neural Networks for Mobile Devices”. CVPR 2016

基于量化的CNN加速与压缩

- 移动设备上的运算效率
  - 数据集：ILSVRC-12

网络	方法	运行时间	硬盘存储	内存占用	Top-5 分类错误率
AlexNet	原始网络	2.93s	232.56MB	264.74MB	19.74%
	Q-CNN^2	0.95s	12.60MB	74.65MB	20.70%
CNN-S	原始网络	10.58s	392.57MB	468.90MB	15.82%
	Q-CNN^2	2.61s	20.13MB	129.49MB	16.68%

Jiaxiang Wu, et al., “Quantized Convolutional Neural Networks for Mobile Devices”. CVPR 2016

大 纲

- 背景介绍
- 排序与索引
- 近似近邻搜索
- **总结与展望**

总结与展望

- 排序与索引是检索和推荐的本质问题
- 正在走向成熟，但依然方兴未艾
- 与其它学科交叉融合

Thanks!

Q&A