CRH: A Simple Benchmark Approach to Continuous Hashing

Miao Cheng

E-mail: mcheng@qdu.edu.cn
Qingdao University
Qingdao, China

IEEE Global Conference of Signal and Information Processing,
Orlando USA, DEC 2015

- Popular of mobile devices.
- Development on information processing.
  - Huge data handling
  - Multimedia applications
  - Human-machine interface
- Ask for efficient solution to data processing.

Background

- To efficiently query, processing, and retrieval huge data
  - Storage
  - Indexing

- Handling of enquiry of data
  - Encode
  - Decode
  - Data matching

- Hashing-based indexing methods
  - Simple ones: Linear
  - Complicated ones: Tree structure, Linear + Tree

Goals

- To design a stable and efficient solution to fast hashing data stream with little system burden as possible.
- Applicable anywhere with real-time feedback.
- Almost no extra requirement.

Most existing methods:

- Handling:
  - Conduct encode and decode procedures separately .
    –> Easily scalable (Almost impractical X)
  - Supervised information is required every once time.
- Solutions:
  - Approximate reconstruction of original data. (V)
  - Discriminative preservation of informative reasons.

Basic Idea

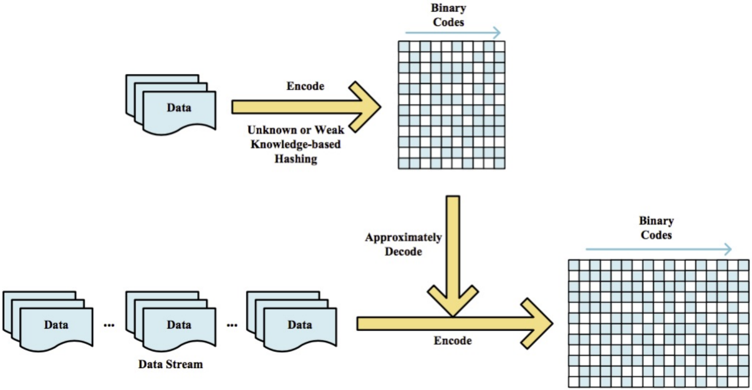
- Random solutions:
  - Fast, stable, and robust under certain conditions.
  - Simple implementation –> small cost

- Random hashing and self-encoder:
  - LSH, RMMH
  - SH, ITQ

- Advantages:
  - Data can be handled independently –> Data stream processing
  - Avoid modification of previous hashing results (usually occurs in tree structures)

Basic Idea

- Encode coming data with a self-adaptive learning of random hashing.



Reconstruction-based (adopted in CRH)

General solutions

- Distance-preserving hashing (V)
- Approximate representation

Reconstruction-based (adopted in CRH)

General solutions

- Distance-preserving hashing (V)
- Approximate representation

Proposed benchmark

- Initial/simple work for further development
- Scalable
- Reasonable

Distances:

- Original data
  - Usually Euclidean distances: $D(x, y)$
- Hashing codes
  - Hamming distances: XOR

Distances:

- Original data
  - Usually Euclidean distances: $D(x, y)$
- Hashing codes
  - Hamming distances: XOR

- Construct bridge between two kinds of distances

- Transform
  - t: 0–1 –> (-1/2)–(1/2)
  - Or alternatives
  - The equivalence of different distances

- Transform
  - t: 0–1 –> (-1/2)–(1/2)
  - Or alternatives
  - The equivalence of different distances

Notice

- Anyway, it hardly works well if whole huge recorded data are referred in calculation
- Feasible only if much limited data is enough

- Transform
  - x, y –> normalized data
  - The equivalence of different distances

- Transform
  - x, y –> normalized data
  - The equivalence of different distances

Notice

- Feasible for originally coming data
- Simple

Objective function

- $Obj\left(s\right) = arg \min_{s \in \{0,1\}} \sum_{i=1}^{q} \sum_{j=1}^{p} \left\| \frac{1}{m} g\left(s_i, t_j\right) - g\left(y_i, x_j\right) \right\|_2$
- where $g\left(\cdot, \cdot\right)$ denotes distances between two data

Comments

- Speciality: kernels (but not equal)
- Further extensions (V)

CRH

- Random selection of referenced data
  - Probability of importance/sampling
  - Lemma: Approximation of a gram matrix
    - Original: different among data
    - CRH: uniform probability

- Random selection of referenced data
  - Probability of importance/sampling
  - Lemma: Approximation of a gram matrix
    - Original: different among data
    - CRH: uniform probability

Results: Random selection is fine ! (V)

- Only a subset data are randomly selected from hashed data associated with binary codes
- Construct formed data with coming data, and calculate low-rank approximate decomposition
- Solve optimization problem

Extensions (Also, possible outlets):

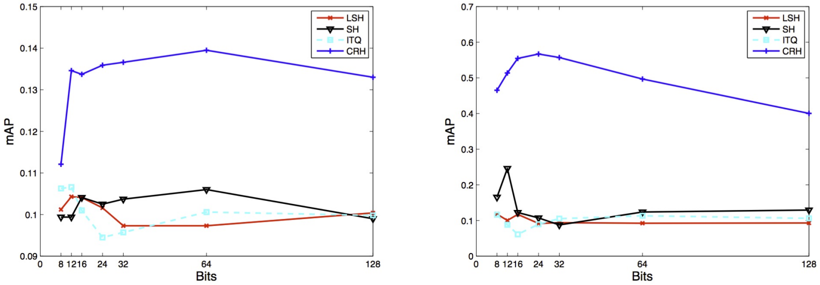
- Add extra regularisation costs into objective
- Discriminative CRH
- Compressive sensing based learning
- Lasso regression

Experiment One:

- Standard hashing
- Data sets: CIFAR-10 and MNIST
- 10000 training vs. 500 testing
  Both randomly selected
- 8-10% data are randomly picked up for encoding

Experimental results

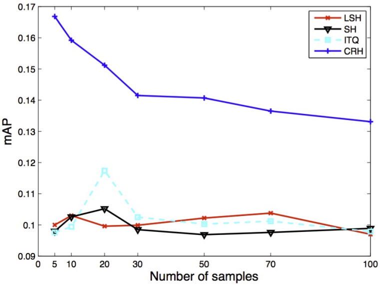
- Different coding bits



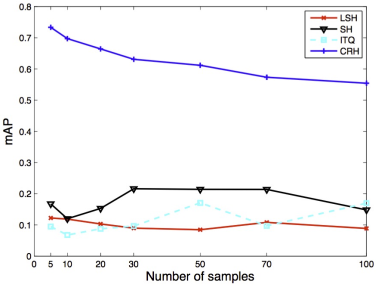
(a) (b)

Figure: The search results from CIFAR-10 and MINIST datasets.

- Different samples in mAP



(a)

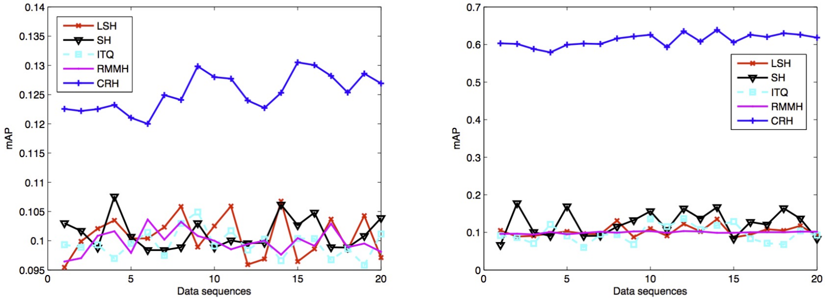


(b)

Figure: The search results from CIFAR-10 and MINIST datasets.

Experiment Two: scalable hashing

- 10000 training data
- Data stream: 500 data sequentally every time



(a) (b)

Figure: The results from CIFAR-10 and MINIST datasets.

Thank you