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
Background

- 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.
Basic Idea

Most existing methods:

- **Handling:**
  - Conduct encode and decode procedures *separately*. 
  - $\Rightarrow$ 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 $\rightarrow$ small cost

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

Advantages:

- Data can be handled \textit{independently} \rightarrow \text{Data stream processing}
- Avoid \textit{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

- $t: 0 \rightarrow (-1/2) \rightarrow (1/2)$
- Or alternatives
- The equivalence of different distances
Transform

- $t: 0-1 \rightarrow (-1/2)-(1/2)$
- 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

- \( x, y \rightarrow \) normalized data
- The equivalence of different distances
Transform
- $x, y \rightarrow$ normalized data
- The equivalence of different distances

Notice
Feasible for originally coming data
Simple
Objective function

\[ \text{Obj}(s) = \arg \min_{s \in \{0,1\}} \sum_{i=1}^{q} \sum_{j=1}^{p} \left\| \frac{1}{m} g(s_i, t_j) - g(y_i, x_j) \right\|_2 \]

where \( g(\cdot, \cdot) \) denotes distances between two data points.

Comments

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

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

- Different samples in mAP

**Figure:** The search results from CIFAR-10 and MINIST datasets.
Experiment Two: scalable hashing

- 10000 training data
- Data stream: 500 data sequentially every time
**Experimental results**

![Graphs showing results from CIFAR-10 and MINIST datasets.](image)

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

Miao Cheng (Qingdao University)
Thank you