Unsupervised Online Hashing

Merits
- label-free, adaptive, space-efficient, single-pass

Categories
- data-independent: Locality-Sensitive Hashing (LSH)
- data-dependent: Online Sketching Hashing (OSH) [1]

This paper focuses on OSH, which has a high accuracy.

OSH and the Problem

OSH: improve the efficiency of PCA-based Hashing via online sketching

1. Sketch \( (A - \mu) \in \mathbb{R}^{n \times d} \) into \( B \in \mathbb{R}^{d \times d} \) with \( B^T B = (A - \mu)^T (A - \mu) \), which requires \( O(nd\ell) \) time and \( O(d\ell) \) space. Here, \( (A - \mu) \) means subtracting each row of the input data \( A \) with its row mean vector \( \mu \in \mathbb{R}^{d \times d} \).

Input: Data matrix \( A \in \mathbb{R}^{n \times d} \), sketching matrix \( B \in \mathbb{R}^{d \times d} \).

1. If \( B \) does not exist then:
   1. Set \( B = 0^{d \times d} \).
   2. For \( i \in [n] \) do:
      1. Insert a \( \alpha_i \) into a zero valued row of \( B \).
      2. If \( B \) has no zero valued rows then:
         1. \( \Sigma = \sum_i B_i \).
         2. Set \( \Sigma = \Sigma V \).
      3. \( \Sigma = \max(\Sigma^T - \sigma^2 I, 0) \).
      4. Set \( B = \Sigma V \).

2. Compute the top \( r \) right eigenvectors of \( B \) denoted by \( W \in \mathbb{R}^{d \times r} \), which takes \( O(d^2\ell^2) \) time and \( O(d\ell) \) space.

3. Run the quantization step as \( h(A) = \text{sgn}((A - \mu)W) \).

Remark. \( \mu \) and \( (A - \mu) \) can be calculated in a streaming fashion with only a single pass, and the details are omitted here. Without sketching, computing the top \( r \) right eigenvectors of \( (A - \mu) \) will take \( O(nd^2) \) time and \( O(nd) \) space. Overall, OSH consumes \( O(nd\ell + d\ell^2) \) time and \( O(d\ell) \) space.

Problem: \( O(n d^2 + d\ell^2) \) time cost is large because \( 1 < \ell < d \).

This Paper: Method and Algorithm

FROSH: reduce the running time of the online sketching

1. Compress the data instances via fast transform.

2. Reduce the variance via distinct fast transforms.

3. Maintain the space efficiency by exploring the properties in the fast transform.

- \( S \in \mathbb{R}^{(\ell/2) \times m} \): sampling matrix
- \( H \in \mathbb{R}^{m \times m} \): Hadamard matrix
- \( H_m = [H_{m/2}, H_{m/2}, -H_{m/2}] \) and \( H_z = [1, 1, 1] \)
- \( D \in \mathbb{R}^{m \times m} \): diagonal Rademacher matrix

Sufficient conditions for the accuracy are:

- \[ \| (A - \mu)^T (A - \mu) - B^T B \|_F^2 \leq \tilde{O} \left( \frac{1}{\ell} + \Gamma(\ell, p, k) \right) \| (A - \mu) \|_F^2 \]

where \( (A - \mu) \in \mathbb{R}^{n \times d} \) means subtracting each row of \( A \) by \( \mu \).

\[ \Gamma(\ell, p, k) = \frac{k}{\sqrt{p\epsilon}} + \sqrt{\frac{k\log 2 + \ell p}{p}} \]

with \( p = \frac{n}{m} \). \( \tilde{O}() \) hides the logarithmic factors on \( (\beta, \delta, k, d, m) \), and the top \( r \) right singular vectors of \( B \in \mathbb{R}^{d \times d} \) are used for hashing projection \( V^T \in \mathbb{R}^{d \times d} \).

The algorithm requires \( O(n \ell^2 + nd + d\ell^2) \) running time with \( O(d\ell) \) space cost after taking \( m = \Theta(d) \).

Remark. \( W \) or its orthogonal rotation from \( B \) is expected to well approximate that from \( (A - \mu) \). Compared with \( O(nd\ell + d\ell^2) \) time cost in OSH, we improve a lot if \( 1 < \ell < d \).