Binary hash functions for fast image retrieval

In K nearest neighbors problem, there are N training points in D-dimensional space (usually D > 100), x ∈ R^D, i = 1, ..., N. and the goal is finding the K nearest neighbors of a query point x ∈ R^D.

Exact search in the original space is O(N^2) in both time and space. A binary hash function h takes as input a high-dimensional vector x ∈ R^D and maps it to an L-bit vector z = h(x) ∈ {0, 1}^L. The search is done in this low-dimensional, binary space.

The main goal is preserving the neighborhood, i.e., assign (dis)similar codes to (dis)similar patterns.

Our hashing model: Binary Autoencoder

We consider binary autoencoders as our hashing model:

\[ h(x) = \sigma(z) \]

where we start with a small \( \mu \) and increase it slowly. To optimize \( \mu \) we apply alternating optimization.

\[ \text{BA} \rightarrow \sum_{n=1}^{N} \left( \sum_{i=1}^{D} (x_{ni} \cdot f(z_{ni}))^2 \right) \]

\[ \text{BA} \rightarrow \sum_{n=1}^{K} \sum_{i=1}^{D} (x_{ni} \cdot g(z_{ni}))^2 \]

where we have \( K \) independent optimizations on \( N \) variables, but it separates into \( N \) independent optimizations each on only \( L \) variables. With \( L \leq 16 \) we can afford an exhaustive search and for larger \( L \), we use alternating optimization.

Advantages of optimizing BA using MAC: It respects the binary constraints and introduces significant parallelism in optimization. Furthermore, the intermediate steps in alternating optimization are (reasonably) easy to solve.

Experiments

Two approaches to initialize \( z_n \) in the Z step:

- Warm start: Initialize \( z_n \) to the code found in the previous iteration \( \mathcal{Z}_{n} \).
- Solve the relaxed problem on \( z_n \) ∈ {0, 1}^L and then truncate it.

The latter achieves better local optima than using warm starts.

We compare our BA that uses a linear hash function and simply minimize the reconstruction error \( E_Z(\mathbf{f}, \mathbf{h}) \) with other methods. To report precision/recall using MAC than using a suboptimal optimization as in ITQ (truncates codes at zero), PCA (finds the best rotation matrix), and sigmoid (relaxes the step function to a sigmoid in training by backpropagation).

The algorithm is highly parallel:

- For fixed \( Z \) we have \( L \) independent problems for each of the \( Z \) single-bit hash functions, and for \( L \).
- For fixed \( Z \) and \( h \) we have \( N \)-independent optimization problems each over \( L \) variables.

Acknowledgements

Work supported by NSF award IIS1423515 and UC Merced’s School of Engineering.