Locality Sensitive Hashing

Ritchie Vink

July 9, 2021

Introduction

Real world use cases

- Recommendation systems
  - Search similar products / users
- De-duplication
  - Determine which website is canonical
- Security
  - Are similar malicious commands executed.

Core problem

All have $O(n)$ search complexity. With the sheer amount of data this is very problematic.

Exact Nearest Neighbor search

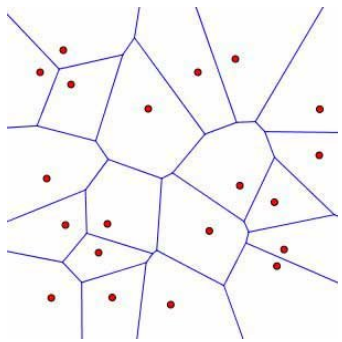
Let:

- P, a set of data point: $p, q \in \mathcal{R}^D$
- $d : \mathcal{R}^D \mapsto \mathcal{R}^1$, a distance function
- $R > 0$, a distance threshold.

Given query q can we find a point $p \in P$, where $d(p, q) \leq R$?

Exact Nearest Neighbor search

- Voronoi cells
  - creation time: $O(n \log n)$
  - query time: $O(\log n)$
  - space: $O(n^D)$
- k-d trees
  - creation time: $O(n \log n$
  - query time in low dimensions: $O(\log n)$
  - query time in high dimensions: $O(n)$
  - space: $O(n)$



Curse of dimensionality

All data structures for exact nearest neighbor search suffer from the curse of dimensionality in time or space complexity.

Approximated Nearest Neighbor search

Relaxed defintion:
The algorithm is allowed to return points $d(p, q) \leq cR$, where $c > 1$.

Locality Sensitive Hashing

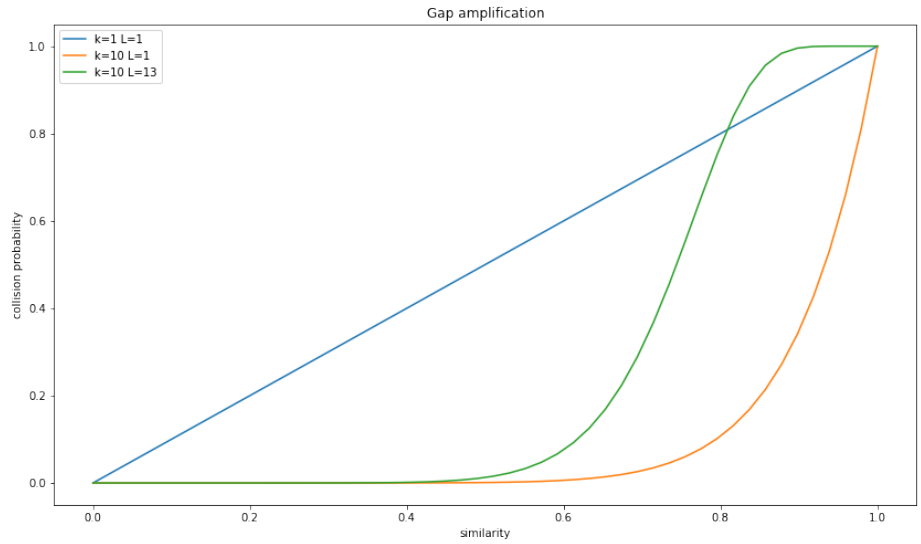
Define a family of hashers where locality is preserved.
let H be a LSH family, such that for every hash function $h \in H$:

- $d(p, q) \leq R$, then $P(h(p) = h(q)) \geq P_1$
- $d(p, q)cR$, then $P(h(p) = h(q)) \leq P_2$

An LSH family is useful when $P(h(p) = h(q)) \leq P_2$.

Gap amplification

- concatenate k hash functions to a single hash (AND operation)
- create L separate hash tables (OR operation)



Locality Sensitive Hashing

Complexity

Given the number of hash digits k, and the number of hashing tables L, and hashing duration h_t we obtain the following complexities:

- creation time: $O(nLkh_t)$
- space: $O(nl)$
- query time: $O(L \cdot (kh_t + DnP_2^k))$

LSH for cosine similarity