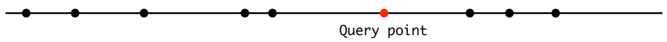
Approximate Near Neighbor Search: Locality Sensitive Hashing

Inge Li Gørtz

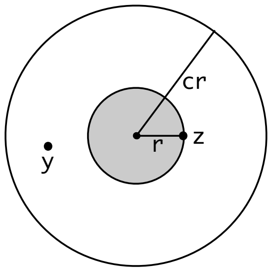
Nearest Neighbor

- Nearest Neighbor. Given a set of points P in a metric space, build a data structure which given a query point x returns the point in P closest to x.
- Metric. Distance function d is a metric:
  1. d(x,y) ≥ 0
  2. d(x,y) = 0 if and only if x = y
  3. d(x,y) = d(y,x)
  4. d(x,y) ≤ d(x,z) + d(z,y)
- Warmup. 1D: Real line



Approximate Near Neighbors

- ApproximateNearNeighbor(x): Return a point y such that $d(x, y) \leq c \cdot \min_{z \in P} d(x, z)$
- c-Approximate r-Near Neighbor: Given a point x if there exists a point z in P $d(x, z) \leq r$ then return a point y such that $d(x, y) \leq c \cdot r$. If no such point z exists return Fail.
- Randomised version: Return such an y with probability δ.



Locality Sensitive Hashing

- Locality sensitive hashing. A family of hash functions H is (r, cr, p_1, p_2)-sensitive with $p_1 > p_2$ and $c > 1$ if:
  - $d(x, y) \leq r \Rightarrow P[h(x) = h(y)] \geq p_1$ (close points)
  - $d(x, y) \geq cr \Rightarrow P[h(x) = h(y)] \leq p_2$ (distant points)

Hamming Distance

- P set of n bit strings each of length d.
- Hamming distance. the number of bits where x and y differ:

$$d(x, y) = |\{i : x_i \neq y_i\}|$$

- Example.

x =	[1]	[0]	1	0	0	1	[0]	0
y =	[0]	[1]	1	0	0	1	[1]	0

Hamming distance = 3

- Hash function. Chose $i \in \{1, \ldots, d\}$ uniformly at random and set $h(x) = x_i$.
- What is the probability that h(x) = h(y)?
  - $d(x, y) \leq r \Rightarrow P[h(x) = h(y)] \geq 1 - r/d$
  - $d(x, y) \geq cr \Rightarrow P[h(x) = h(y)] \leq 1 - cr/d$

LSH with Hamming Distance

- Pick k random indexes uniformly and independently with replacement. Let
  - $g(x) = x_{i_1} x_{i_2} \cdots x_{i_k}$
- Probability that g(x) = g(y)?
  - $d(x, y) \leq r \Rightarrow P[g(x) = g(y)] \geq (1 - r/d)^k$
  - $d(x, y) \geq cr \Rightarrow P[h(x) = h(y)] \leq (1 - cr/d)^k$
- Bucket: Strings with same hash value g(x).
- LSH:
  - Construct L hash tables L_j each with its own independently chosen function g_j.
  - Insert(x): Insert x in the list of $g_j(x)$ in L_j.
  - Query: For all $1 \leq j \leq k$ check each element in bucket $g_j(x)$ in L_j.

LSH with Hamming Distance

Let $k = \frac{\log n}{\log(1/p_2)}$, $\rho = \frac{\log(1/p_1)}{\log(1/p_2)}$, and $L = \lceil 2n^\rho \rceil$, where $p_1 = 1 - r/d$ and $p_2 = 1 - cr/d$.

- Claim 1. *If there exists a string z* in P with d(x,z*) ≤ r then with probability at least 5/6 we will return some y in P for which d(x,y) ≤ cr.*
- Probability that z* collides with x:
  - $P[\exists \ell : g_\ell(x) = g_\ell(z^*)] = 1 - P[g_\ell(x) \neq g_\ell(z^*) \text{ for all } \ell]$

$$= 1 - \prod_{\ell=1}^{L} P[g_\ell(x) \neq g_\ell(z^*)]$$

$$= 1 - \prod_{\ell=1}^{L} \left(1 - P[g_\ell(x) = g_\ell(z^*)]\right)$$

$$\geq 1 - \prod_{\ell=1}^{L} (1 - p_1^k) = 1 - (1 - p_1^k)^L \geq 1 - e^{-Lp_1^k}$$

$$\geq 1 - \frac{1}{e^2} \geq 1 - 1/6 = 5/6$$