6.851: Advanced Data Structures Spring 2007

Lecture 10 — March 14, 2007

Alex Andoni *Scribe: Alex Schwendner*

1 Nearest Neighbor

In the Nearest Neighbor problem, we are given a set P of points in $\mathbb{R}^d$ or another metric space. We want to construct a data structure such that we can efficiently answer queries in which we are given a point q and must return a point $p \in P$ minimizing the distance $\|p - q\|$ between q and p.

1.1 Interesting Metrics

- **Euclidean Distance** on $\mathbb{R}^d$

$$\|p - q\|_2 = \sqrt{\sum_{i=1}^{d} |p_i - q_i|^2}$$

- **Manhattan** on $\mathbb{R}^d$

$$\|p - q\|_1 = \sum_{i=1}^{d} |p_i - q_i|$$

- **Hamming** on $\{0, 1\}^d$

$$\|p - q\|_1 = \sum_{i=1}^{d} |p_i - q_i|$$

  The Hamming distance is the same as Manhattan distance except only on $\{0, 1\}$.

2 Applications

Similarity search problems can often be expressed in this form with a suitable metric.

2.1 Inexact String Matching

Here, our metric is the number of differences between two strings p and q (essentially the Hamming metric). If we have a text T and a pattern P, then we consider all $|T| - |P| + 1$ length-$|P|$ substrings of T. Our goal is then to find the closest such point to P.

2.2 Image Matching

If we have an image, we can consider each pixel as a coordinate. Then, the images is a high dimensional vector. We can use the Manhattan metric to express the distance between two images. Another way we might obtain a metric is to compute a Fourier transform of the two images and then define the distance to be the Manhattan distance between the Fourier coefficients.

3 Exact Algorithms

3.1 $\mathbb{R}^2$: Voronoi Diagrams

In two dimensions, the nearest neighbor problem can be solved with a Voronoi diagram. The Voronoi diagram on a set P of n points in the plane is a partition of the plane into n *Voronoi cells*. Each Voronoi cell is the set of points closer to some specified point $p \in P$ than to any other point in P. Each Voronoi cell is a (possibly unbounded) region with straight sides.

The construction of a representation of a Voronoi diagram can be done in $O(n \log n)$ time. The Voronoi diagram takes $O(n)$ space and with it, nearest neighbor queries can be answered in $O(\log n)$ time.

3.2 Higher Dimensions

While constructing a Voronoi diagram nicely solves the two dimensional problem, the algorithm does not extend well to higher dimensions. Analogous Voronoi structures on n points in d dimensions use $n^{O(d)}$ space.

The naive algorithm of simply computing the distance from a query to each of the n points takes $O(dn)$ time for each query.

The difficulty of exactly computing nearest neighbors in d dimensions is the reason for considering the problem of computing approximate nearest neighbors.

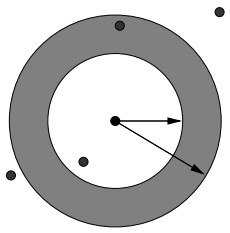
4 Approximate Near Neighbor

4.1 Near Neighbor

Near Neighbor is the decision version of Nearest Neighbor. Given a set P of points in $\mathbb{R}^d$ and a real number $r > 0$, we want to construct a data structure such that we can efficiently answer queries in which we are given a point q and must return some point $p \in P$ such that $\|p - q\| \leq r$, if such a point exists.

4.2 c-Approximate Near Neighbor

For the *c-Approximate Near Neighbor* problem, we want to construct a data structure such that when given a query q for which there exists a point $p \in P$ with $\|p - q\| \leq r$ we can quickly find some point $p' \in P$ with $\|p' - q\| \leq cr$.



Given a c-approximate algorithm for Near Neighbor, we can construct a c^2-approximate algorithm for Near*est* Neighbor. Suppose all distances between points are in the range $[1, \Delta]$. Then, we can use binary search.

Let D_r be the c-approximate Near Neighbor algorithm with distance cutoff r. Suppose we are given input point q. Then, consider the sequence

$$\underbrace{D_1, D_c, D_{c^2}, \ldots, D_\Delta}_{\log \Delta}.$$

We binary search within this sequence to find the smallest i for which $D_{c^i}(q)$ returns a valid point. We then return this as our answer, which will be within a factor of c^2 of optimal.

5 Locality-Sensitive Hashing

Here, our approach will be to devise a weak locality-sensitive hash function and amplify it to a stronger one. The details of the primitive hash function will depend on the metric and space used. We shall first present how to perform the amplification, and then discuss some particular weak hash functions.

5.1 Amplification

These results are by Piotr Indyk and Rajeev Motwani in [IM98].

The idea is to solve c-approximate Near Neighbor with hashing. Loosely speaking, we want to construct hash functions $g \colon \mathbb{R}^d \longrightarrow U$ such that

- if $\|p - q\| \leq r$, then $\Pr[g(p) = g(q)]$ is not too small,
- if $\|p - q\| > cr$, then $\Pr[g(p) = g(q)]$ is small.

More formally, a family H of functions $h \colon \mathbb{R}^d \longrightarrow U$ is called (P_1, P_2, r, cr)-sensitive if for any p, q

- if $\|p-q\| \leq r$, then $\Pr[h(p)=h(q)] > P_1$,
- if $\|p-q\| > cr$, then $\Pr[h(p)=h(q)] < P_2$.

We now amplify the gap between P_1 and P_2 by concatenating $k = \log_{P_1/P_2} 2n$ different such hash functions to get a new hash function h'. Then,

- if $\|p-q\| \leq r$, then $\Pr[h'(p)=h'(q)] > P_1^{\log_{P_1/P_2} 2n} = 2n \cdot P_2^{\log_{P_1/P_2} 2n}$,
- if $\|p-q\| > cr$, then $\Pr[h'(p)=h'(q)] < P_2^{\log_{P_1/P_2} 2n}$.

If we now consider n^ρ such amplified hash functions for $\rho = \frac{\log P_1}{\log(P_2/P_1)}$, then the chance of any collision will be $O(1)$ if $\|p-q\| \leq r$ and small if $\|p-q\| > cr$.

This yields query time of $O(dn^\rho \log n)$ and space of $O(n^{\rho+1} + dn)$.

5.2 LSH for Particular Metrics

- **Euclidean**

  Here, divide space into hyper-boxes with edge length $r(1+\epsilon)$, and hash each point to the box containing it. If $\|p-q\| \leq r$, then $\Pr[h(p)=h(q)] \geq \frac{\epsilon}{1+\epsilon}$. If $\|p-q\| > r\sqrt{d}$, then $\Pr[h(p)=h(q)] = 0$.

- **Hamming**

  Let our family H of hash functions be $\{h_i(p) = p_i\}$ where p_i is the i-th bit of p. Then, $\Pr[h(p)=h(q)] = 1 - \frac{1}{d}D(p,q)$ where $D(p,q)$ is the Hamming distance between p and q.

References

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, New York, NY, USA, 1998. ACM Press.