Seminar on Algorithms and Geometry 2014B
Lecture 1 – Doubling metrics and Nearest Neighbor Search*

Robert Krauthgamer

1 Introduction

Algorithms meet geometry?:

I interpret geometry as anything that involves distances (metric spaces), including Euclidean, other norms, tree metrics, edit distances, earthmover distance, etc.

These concepts arise in many algorithmic settings: as part of the model/problem (e.g. the input is points in the plane) or from the algoithmic method of solving it (e.g. for cut problems, we use a linear program that "produced" a metric).

Motivation:

Use geometric tools (mathematics) to design good algorithms

2 Nearest Neighbor Search (NNS)

Setup: a metric space (M, d)

Examples: Euclidean space R^k, or a collection of DNA sequences and the edit distance between them

Assume for simplicity that $d(x, y)$ can be computed in $O(1)$ time

Problem definition:

Preprocess: a collection (database) of n points $S \subset M$

Query: Given a point $q \in M$, find its closest data, i.e., $a \in S$ that minimizes $d(q, a)$.

Naive solution: no real preprocessing (just store the points in $O(n)$ space), and at query time search S exhaustively (by n distance computations) in $O(n)$ time.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

Holy grail: preprocessing $O(n)$ and query time $O(\log n)$ [sorting reals, in dimension one]

A lower bound under black-box access:

Assume only black-box access to the distance between points.

Lemma: There is a dataset that requires (worst-case) $\Omega(n)$ distance computation to answer an NNS query, even with preprocessing.

Idea: S is a large uniform metric

Question: Is this the "only" obstruction to fast algorithms?

This is a "high-dimensional" phenomenon. How can we exclude this scenario?

3 Doubling metrics

Defn: A ball $B(x,r) := \{y \in M : d(x,y) \le r\}$.

Defn: The doubling dimension of a metric space (M,d) is the smallest $k > 0$ such that every ball can be covered by at most 2^k balls of half the radius. We denote it $\text{ddim}(M)$.

Exer: Prove that the doubling dimension of k-dimensional Euclidean space is $O(k)$. And the same for ℓ_∞-norm.

Exer: Let $k = \text{ddim}(M)$ and define k' similarly using diameter instead of radius (covering by sets of half the diameter). Prove that $k' = \Theta(k)$.

Exer: Suppose $M = M_1 \cup M_2$. Prove that $\text{ddim}(M) \le O(\text{ddim}(M_1) + \text{ddim}(M_2))$.

Exer: Let $M' \subset M$ be a submetric of (M,d). Prove that $\text{ddim}\, M' \le O(\text{ddim}\, M)$.

Exer: Let M contain all vectors in $\mathbb{R}^m$ that are k-sparse (have at most k nonzeros), and let d be the Euclidean distance (ℓ_2-norm). Prove that (M,d) has doubling dimension $O(k \log m)$.

Defn: The aspect ratio (or spread) of S is $\Phi(S) := \frac{\max_{x,y\in S} d(x,y)}{\min_{x\neq y\in S} d(x,y)}$.

(We assume throughout all distances are strictly positive.)

Packing Lemma: Let $S \subset M$ be finite. Then

$$|S| \le (4\Phi(S))^{\text{ddim}(M)}.$$

Conclusion: A metric of low doubling dimesion does not have a large (near) uniform metric.

Proof: Seen in class.

4 Nets

Will take the role of "grids" (of some resolution) in Euclidean spaces.

Defn: An r-net of M is a subset $Y \subset M$ satisfying

1. Packing: for all distinct $y, y' \in Y$ we have $d(y, y') > r$;
2. Covering: for all $x \in M$ we have $d(x, Y) = \min_{y \in Y} d(x, y) \leq r$.

Greedy construction of nets: Find a point that is not currently covered and add it to Y, and repeat

More formally: Initialize $Y = \emptyset$, and iterate over all points $x \in M$, and if this x is not covered by the current Y, just add it to Y.

5 NNS in doubling spaces

We decribe an scheme for $(1+\varepsilon)$-approximate NNS, i.e., report a point a such that

$$d(a, q) \leq (1+\varepsilon) \min_{x \in S} d(x, q).$$

Theorem: One can preprocess a subset $S \subset M$ of size n, and build a data structure of size $2^{O(\operatorname{ddim} S)} \cdot n$, so as to answer $(1+\varepsilon)$-NNS queries (for every $\varepsilon < 1/2$) in time $(1/\varepsilon)^{O(\operatorname{ddim} S)} \cdot \log \Phi(S)$.

Assume by normalization that $\min_{x \neq y \in S} d(x, y) = 2$.

Remark: Can do also insertion and deletion (updates to the set S) in similar time $2^{O(\operatorname{ddim} S)} \cdot \log \Phi(S)$.

Remark: There are subsequent refinements, like replacing $\log \Phi(S)$ with $\log n$, or (alternatively) improving the space to $O(n)$, but it is sometimes on the expense of simplicity.

Preprocessing procedure: For every integer i from 0 to $m := \lceil \log_2 \operatorname{diam}(S) \rceil$ construct a 2^i-net of S, called Y_i.

Observe that $Y_0 = S$ and $|Y_m| = 1$.

We can further ensure the nets are nested, i.e., each $Y_i \subset Y_{i-1}$. How? In the greedy construction of Y_{i-1}, the order is arbitrary so if we start with the points of Y_i, these points will surely be included.

For every point level i and $y \in Y_i$, construct a list ("pointers" to nearby lower-level net-points)

$$L_{y,i} = \{z \in Y_{i-1} : \ d(y, z) \leq 3 \cdot 2^i\}.$$

The packing lemma immediately implies that $|L_{y,i}| \leq 2^{O(\operatorname{ddim} S)}$.

Preprocessing space: We can bound it by $\sum_i \sum_{y \in Y_i} |L_{y,i}| \leq 2^{O(\operatorname{ddim} S)} n \log \Phi(S)$. We will later show an improved analysis that uses the nesting.