6.838: Geometric Computing

Spring 2005
Problem Set 3
Solutions

Problem 1. Fast Approximate Near Neighbor in l_1

Construct a new data structure for the Approximate Near Neighbor problem in $\Re^d$ under l_1 norm. Your data structure should have the following parameters:

- Approximation factor: $O(d)$
- Space: $O(dn)$
- Query time: $O(d)$

Your data structure can be randomized.

Solution: We assume that the parameter $r > 0$ of the near neighbor problem is equal to 1. The main idea is to use a randomly shifted grid. That is, let G be a square grid in $\Re^d$, with cell side length 2. The grid is shifted a a vector t chosen uniformly at random from $[0, 2]^d$. For any point $p \in \Re^d$, we use $G(p)$ to denote the grid cell containing p.

Denote the set of n input points by P. During the preprocessing, the algorithm stores each point $p \in P$ in a bucket $G(p)$ (this can be implemented using either a hash table, or trees). Given a query point q, the algorithm checks if $G(q)$ is non-empty. If so, it reports any point from $G(q)$. Observe that any point in $G(q)$ is within distance $2d$ from q.

The query time is $O(d)$, the space is $O(dn)$. It suffices to show that if there is a point $p \in P$ such that $\|p-q\|_1 \leq 1$, then $\Pr[G(p) = G(q)] \geq 1/2$. To this end, observe that if $G(p) \neq G(q)$, then in one dimension $i = 1 \ldots d$ we must have $\lfloor (p_i - t_i)/2 \rfloor \neq \lfloor (q_i - t_i)/2 \rfloor$. Each such event happens with probability $|p_i - q_i|/2$. Thus

$$\Pr[G(p) \neq G(q)] \leq \frac{\sum_i |p_i - q_i|}{2} = \|p - q\|_1/2 \leq 1/2$$

As a side comment, observe that the probability of failure can be easily made, say, at most $1/2^3 = 1/8$, e.g., by building 3 data structures using independent random bits, and querying all 3 of the of them.

Problem 2. LSH in $\Re^d$ under l_1

In the class, we have seen how to embed $\{0 \ldots M\}^d$ equipped with l_1 norm, into the Hamming space $\{0, 1\}^{Md}$. This automatically yields a randomized data structure solving a c-approximate Near Neighbor with query time $O(dMn^{1/c})$, for $c = 1 + \epsilon > 1$.

Show how to extend the latter data structure so that it works for points in $\Re^d$ (again, the distance is defined by the l_1 norm). Your data structure should support queries in time $O((d \log n/\epsilon)^{O(1)} n^{1/c})$.

Solution: In the first step, we apply the preprocessing algorithm from Problem 1 to partition the input set P into sets $P(a) = P \cap a$, for all cells $a \in G$. For each $P(a)$ we will construct a separate NN data structure. Given a query q, we will query the data structure constructed for the set $P(a)$ for $a = G(q)$.

By the analysis in Problem 1, we know that for each q and p such that $\|q - p\|_1 \leq 1$, we have $G(p) = G(q)$ with high constant probability . At the same time, we know that $P(a)$ is contained in a cube of side 2. For simplicity we assume $P(a) \subset [0, 2]^d$.

To solve the problem for $P(a)$, we discretize the points' coordinates. For an integer $k \geq 1$ let $r_k(x) = \lfloor x * k \rfloor / k$ (we specify k later). For a point $p = (p_1 \dots p_d) \in \Re^d$, $r_k(p) = (r_k(p_1) \dots r_k(p_d))$. During the preprocessing, we replace each point $p \in P(a)$ by $r_k(p)$; call the resulting set $r_k(P)$. Since the coordinates of points in $r_k(P)$ are in the range $\{0, 1/k, \dots, 2 - 1/k, 2\}$, we can use the LSH algorithm from the lecture. Specifically, by scaling, we can assume that the coordinates are in the range $\{0 \dots M\}$ for $M = 2k$. Then, we build a c'-approximate near neighbor data structure for $r_k(P)$, for some $c' < c$ and radius r' specified later. It uses $O(dn^{1+1/c'})$ space and has $O(Mdn^{1/c'})$ query time. Given a query point q, we query the data structure with the point $r_k(q)$.

It remains to specify the values k, c' and r', so that (a) the approximation factor of the algorithm is $\leq c$ and (b) the exponent in the query time and space bound is $1/c$. We set $k = d \log n$. In this way we ensure that $\|r_k(p) - r_k(q)\|_1 - \|p - q\|_1 \leq 1/\log n$. Thus, if $\|p - q\|_1 \leq 1$, then $\|r_k(p) - r_k(q)\|_1 \leq 1 + 1/\log n$. We set $r' = 1 + 1/\log n$. The c'-approximate NN data structure can return a point whose distance from q is at most $c'r'$. We let $c'r' = cr = c$ by setting $c' = c/r' = c/(1+1 \log n)$. In this way, we ensure that the data structure is correct (with constant probability).

What the query time of the algorithm ? It is

$$O(Mdn^{1/c'}) = O(d^2 \log n \cdot n^{(1+1/\log n)/c}) = O(d^2 \log n \cdot n^{1/c} n^{1/\log n}) = O(d^2 \log n \cdot n^{1/c})$$

The space bound is similar.

Note: there is an alternative solution to this problem, that does not use embedding into Hamming space. Its main idea is to show that if the cell side length in the grid G is large enough, then $G(p)$ itself is a "good" locality-sensitive hash function. If you are interested in details of this solution, ask me about it during office hours.

Problem 3. $(1, 2) - B$ **metrics**

In the class, we have seen how to construct an exact embedding of a given metric $M = (X, D)$, $|X| = n$, into l_∞^n. In this problem we consider embeddings of a special subclass of metrics called $(1, 2) - B$ *metrics*. A metric is a $(1, 2) - B$ metric if it satisfies the following two very particular conditions:

1. All non-zero distances are either 1 or 2
2. For any point $p \in X$, the number of points $q \in X$ such that $D(p, q) = 1$ is at most B.

Show that there is a constant C such that any metric M satisfying the above conditions can be embedded exactly into l_∞^d where $d = CB \log n$.

Solution: See V. Guruswami, P. Indyk, "Embeddings and Non-approximability of Geometric

Problems", Proceedings of the 14th Symposium on Discrete Algorithms, 2003.
At http://theory.csail.mit.edu/~indyk/kmed2.ps.