Nearest Neighbor Searching (NNS)

$S$: a set of $n$ points in $\mathbb{R}^2$.

$q$: any query point $\mathbb{R}^2$.

Find its nearest neighbor $p^* \in S$ satisfying $d(q, p^*) = \min_{p \in S} d(q, p)$.

Applications

- Pattern Recognition, Data Compression
- Statistical Classification, Clustering
- Databases, Information Retrieval
- Computer Vision, etc.

http://en.wikipedia.org/wiki/Nearest_neighbor_search
$S = \{p_1, p_2, \cdots, p_n\}$, a set of $n$ points in $\mathbb{R}^2$

Voronoi cell of $p_i$:

$$\mathcal{V}(p_i) = \{q \mid d(q, p_i) \leq d(q, p_j), \forall 1 \leq j \leq n, j \neq i\}$$

$\text{Vor}(S)$ offers an optimal solution:

preprocessing: $O(n \log n)$

space: $O(n)$

query: $O(\log n)$

Nearest Neighbor Searching
In High Dimensions

**Exact Algorithms**

- Logarithmic query time -> (roughly) $n^{d/2}$
- (Near-)linear space, best bound for *random* input: $\min(2^{O(d)}, dn)$
- Best known method, $O((n/m^{1/\lceil d/2 \rceil}) \polylog n)$ query time, $O(m)$ space, $(n < m < n^{\lceil d/2 \rceil})$. [Agarwal et al. 1993]

**Approximation Algorithms**

- For fixed $d$, optimal for $(1 + \varepsilon)$- approximate NN. However, actual query time: $O((1/\varepsilon)^d \log n)$.[Arya et al. 1994]
- Near-optimal hashing algorithms based on locality-sensitive hashing. [Indyk et al. 2008]
Inherent in many applications, *e.g.*, measurement errors in sensor data.

- **Locational model**
- **Existential model**
- **Deterministic model**

- Minimum spanning trees [Suri et al. 2011]
- Closest pair and the post office problem [Suri et al. 2011]
Nearest Neighbor Searching Under Uncertainty

$S = \{P_1, P_2, \cdots, P_n\}$, a set of uncertain points in $\mathbb{R}^2$.

$P_i$, represented by a probability density function (pdf).

**Discrete pdf**

- $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$

**Continuous pdf**

- Gaussian Distribution
Nearest Neighbor In Expectation

$q$: query point

Expected distance from $q$ to $P_i$: \[ Ed(q, P_i) = \sum_{j=1}^{k} w_{i,j} d(q, p_{i,j}). \]

Problem: find its expected nearest neighbor $P^*$,

\[ Ed(q, P^*) = \min_{P \in S} Ed(q, P). \]
Nearest Neighbor With Highest Prob

$q$: query point

$\varphi(q, p_{i,j})$: Prob that $p_{i,j}$ is NN of $q$

$$
\varphi(q, p_{i,j}) = \prod_{1 \leq t \leq n, t \neq i} w_{i,j} \left( 1 - \sum_{u_t \in \vartheta(u_t)} w_{t,u_t} \right),
$$

where $\vartheta(u_t) = \{v \mid d(q, p_{t,v}) < d(q, p_{i,j})\}$.

Prob that $P_i$ is NN of $q$: $\Phi(q, P_i) = \sum_{j=1}^{k} \varphi(q, p_{i,j})$.

**Problem:** find its nearest neighbor $P^*$ with highest probability,

$$
\Phi(q, P^*) = \max_{P \in S} \Phi(q, P)
$$
Proposed Approaches: Nearest Neighbor In Expectation

Expected Voronoi cell of $P_i$:

$$\mathcal{EV}(P_i) = \{ q \mid Ed(q, P_i) \leq Ed(q, P_j), \forall 1 \leq j \leq n, j \neq i \}.$$ 

i.e., set of points for which $P_i$ is expected NN.

Expected Voronoi diagram $EVor(S)$: a collection of expected Voronoi cells.

Unfortunately, the bisector can be a complex curve.

For Gaussian distribution, bisector is a line!

$$4x^3 + 12x^2y - 49x^2 + 4xy^2 - 94xy + 196x + 12y^3 - 65y^2 + 236y - 272 = 0$$
Moreover, the expected Voronoi cell can be disconnected.
Future work:

- Study the complexity of expected Voronoi diagram
- Do some experiments
- approximate the expected Voronoi diagram
- Approximate directly

Given linear space, can we achieve $O(\sqrt{n})$ query time?
Thanks!

Questions?