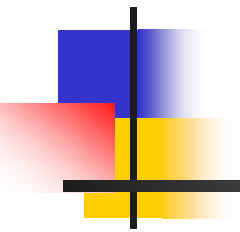
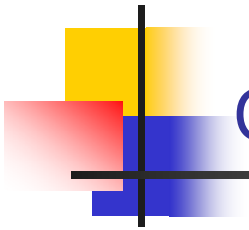


Nearest Neighbor Searching and Metric Space Dimensions

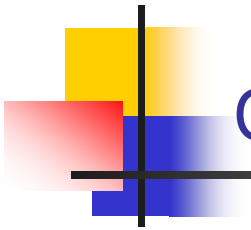
Presented by: Nakul Verma

Feb 8, 2007.



Outline

- Motivation
- Introduction to various concepts of dimensions
- Applications to Nearest Neighbor Queries

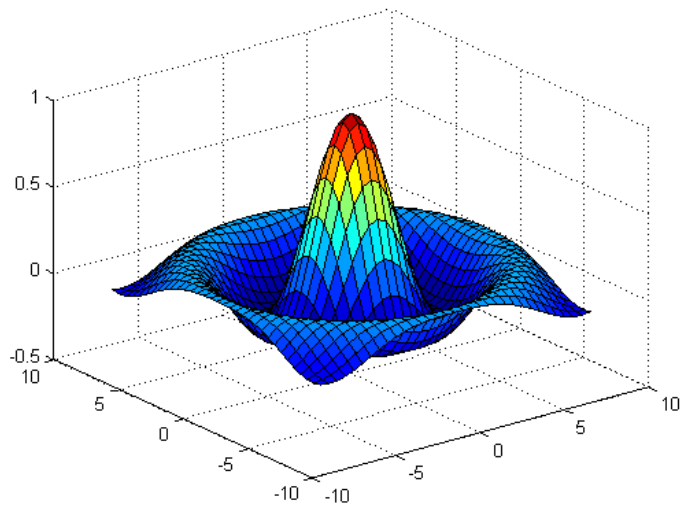


Outline

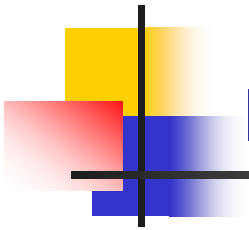
- Motivation
- Introduction to various concepts of dimensions
- Applications to Nearest Neighbor Queries

Why need a new concept of dimension?

- We all know that many algorithms scale poorly with dimensionality of the data.
- Researchers working on manifolds have been able to come up with algorithms which are only dependent on the dimensionality of the manifold.
- So how can we formalize the concept of ‘intrinsic’ dimension?



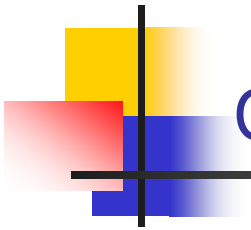
even though, the surface resides in R^3, its effective dimension is 2



Idea

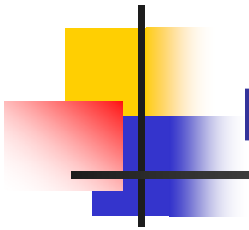
- Consider a rectangular segment in R^2, let $\varepsilon > 0$ be small, then what is the minimum number of balls of radius ε are needed to cover this rectangular segment? ~ $(1/\varepsilon)^2$
- Similarly for a line segment this quantity is ~ $(1/\varepsilon)$
- Observations
  - The scaling behavior is not a function of the underlying space!

- Variants of the above idea have been formalized and well studied. Moreover, they are particularly effective in speeding up some algorithms!



Outline

- Motivation
- Introduction to various concepts of dimensions
- Applications to Nearest Neighbor Queries



Notions of dimension

- We can characterize dimension using many methods
  - Box dimension
  - Hausdorff dimension
  - Pointwise dimension
  - and many more !
  - Assouad dimension
  - Doubling dimension
  - Doubling measure
- Useful notation:
  - U – set [Universe], $S \subseteq U$
  - D(.,.) – distance function
  - $B(x, \varepsilon) = \{y \in U \mid D(x,y) \leq \varepsilon \}$ [Ball centered at x of radius ε]
  - C(S, ε) – Covering number: size of the smallest ε covering of set S

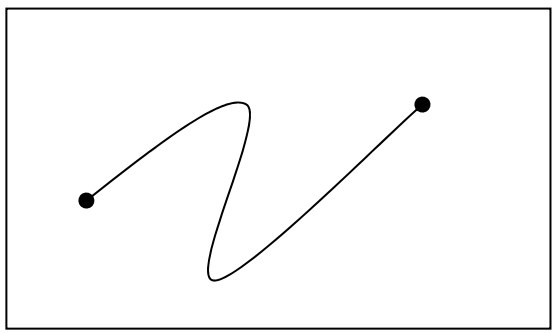
Box dimension

- Definition: The box dimension $\dim_B(Z)$ of metric space $Z = (U,D)$ is the d such that the covering number satisfies:

$$\mathbf{C}(\mathbf{U}, \varepsilon) = 1/\varepsilon^{d+o(1)} \qquad \text{as } \varepsilon \to 0$$

- Some examples:

Line segment in $\Re^2$



Box Dim = 1

Cantor set in $\Re$ [0,1]

Box Dim = $\log_3 2 \approx 0.63$

Assouad dimension

- Related to Box dimension
- Definition: The value d (if it exists) such that

$$\sup_{x \in \mathbf{U},\, r>0} \mathbf{C}(B(x,r), \varepsilon r) = 1/\varepsilon^{d+o(1)} \qquad \text{as } \varepsilon \to 0$$

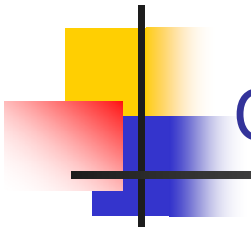
- Observations:
  - Stronger condition than box dimension.
  - Take ε = ½, and we have that any ball B(x,r) is contained in the union of at most 2^d balls of radius r/2. Thus d is termed as the *doubling dimension*.
  - Unlike box dimension, can assign d > 0 to a finite point set.

Doubling Measure

- If Z is a metric measure space (U,d,μ)
- Definition: The value d such that, for all x, r > 0

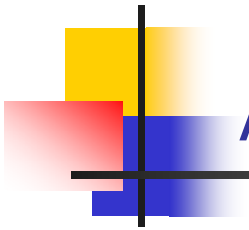
$$\mu(B(x,2r)) \leq \mu(B(x,r))2^d$$

- Observations:
  - Can be thought as the 'measured' version of Assaouad dimension



Outline

- Motivation
- Introduction to various concepts of dimensions
- Applications to Nearest Neighbor Queries



Application: Nearest Neighbors (NN)

- Traditionally for a query point $q \in U$, and sites $S \subset U$, s.t. $|S|=n$, NN search takes time and space $O(nd)$
- Assuming that data has a bounded dimension, we can perform faster (approx) NN queries.
- We will explore a Divide and Conquer technique for fast NN in following contexts:
  - Data resides in a space having a doubling constant
  - Data resides in a space having a doubling measure

Nearest Neighbors (with doubling constant)

- Let Z=(U,D) be the metric space with bounded doubling dimension d, $S \subset U$ be the *input sites*, $q \in U$ be the query point.
- Goal: to find a point $a \in S$, s.t. for all $x \in S$
  - $D(a,q) \leq D(x,q)$ [well... approximately]

- *Theorem:* If we scale distances s.t. all sites in S are at most 1 apart, then we can find an approx NN of q in S in

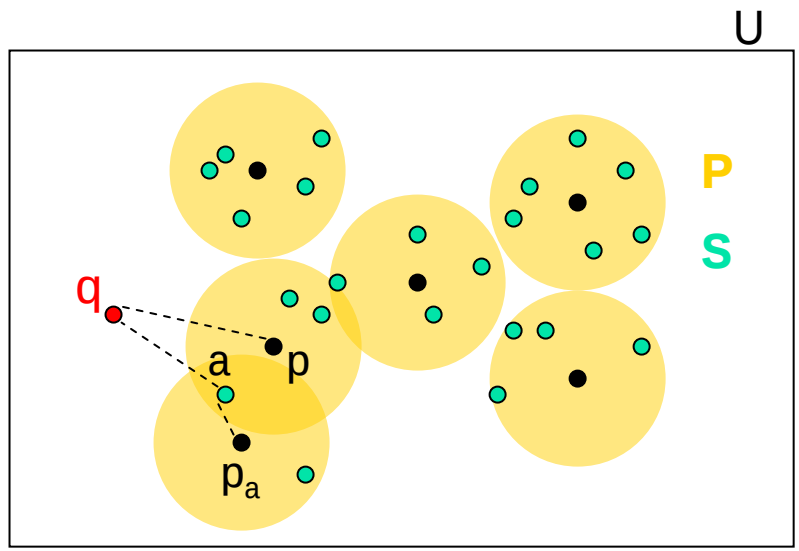
$$O\left(2^{O(d)} \log \Delta(S)\right)$$

where $\Delta(S)$ is the ratio of the largest distance to the smallest distance of points in S

Proof

- Idea: create an epsilon net!
  - Let P be a δ^2-net of S, for some $\delta > 0$, then $|P| \leq O(1/\delta^{2d})$ [doubling space]
    Let $a \in S$ be closest to q (in S), $p \in P$ be closest to q (in P), $p_a \in P$ be closest to a (in P)
  - Note that: $D(q,p) \leq D(q,p_a) \leq D(q,a) + D(a,p_a) \leq D(q,a) + \delta^2$
  - If $D(q,a) > \delta$, we immediately have that p is $(1+\delta)$-near to q in S.
  - If $D(q,a) \leq \delta$, we have
    $D(p,a) \leq D(p,q) + D(q,a) \leq 2\delta + \delta^2 \leq 3\delta$

    ie, we can confine our answer to ball: $B(p,3\delta)$



Proof (cont.)

- Gives a natural recursive datastructure to answer the query
  - Recursively build datastructure, for each $p \in P$, for $S \cap B(p,3\delta)$
  - If we make sure that at depth t, the relevant sites are in a ball of radius $1/2^t$, we'll have that:
  - the depth is proportional to $\log(\Delta(S))$
  - => NN in time $O(2^{O(d)} \log(\Delta(S)))$

- Observations:
  - We can answer the query approximately in time independent of |S|
  - We somehow need to construct an epsilon-net
  - Storage required is superlinear ☹. More efficient datastructures are considered by Krauthgamer and Lee (2004)

Nearest Neighbors (with doubling measure)

- Let $Z=(U,D,\mu)$ be a metric measure space with doubling measure d, $S \subset U$ be the *input sites*, $q \in U$ be the query point. $|S|=n$

- Idea: to get iid samples from S and will conclude that with high probability we can confine our search to a fraction of original points in S. Then recurse!

- *Theorem:* Suppose P is a random subset of S, where $x \in S$ is chosen independently with probability m/n. Let p be the closest point to q (in P). Then with probability at least $1-1/n^{K/4^d}$, the nearest neighbor to q in S will be contained in a ball around p of size Kn(log n)/m

Proof

- Note: abbreviating $|S \cap B(x,r)|$ as $|B(x,r)|$,
  m and K determined later
- For each $p \in P$, consider ε_p such that $|B(p,\varepsilon_p)| = Kn(\log n)/m$
- [good event] For the nearest p to q (in P), if $D(q,p) \leq \varepsilon_p/2$, then nearest site to q (in S) is contained in $B(p,\varepsilon_p)$
  - Because: $D(p,a) \geq 2D(p,q) \rightarrow D(q,a) \geq D(p,q)$

    [$D(p,a) \geq D(p,q)+D(p,q) \geq D(p,a)-D(a,q)+D(p,q)$, refer figure below]

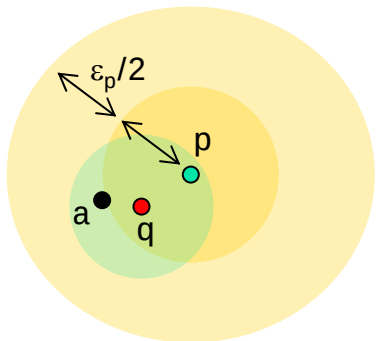


Figure: $q \in B(p,\varepsilon_p/2) \rightarrow a \in B(p,\varepsilon_p)$

Proof (cont.)

- [bad event] If $\beta := D(q,p) > \varepsilon_p/2$, then

$$|B(q,\beta)| \geq |B(q,3\beta)|/4^d \geq |B(p,\varepsilon_p)|/4^d = Kn(\log n)/m4^d$$

[for $x \in B(p,\varepsilon_p)$ - $D(q,x) \leq D(q,p)+D(p,x) \leq \beta + \varepsilon_p \leq 3\beta$, refer figure below]

$\therefore$ Pr[p closest to q (in P)]

$\leq$ Pr[B(q,β) doesn't any points of P]

$\leq (1-m/n)^{Kn(\log n)/m4^d}$

$\leq 1/n^{K/4^d}$

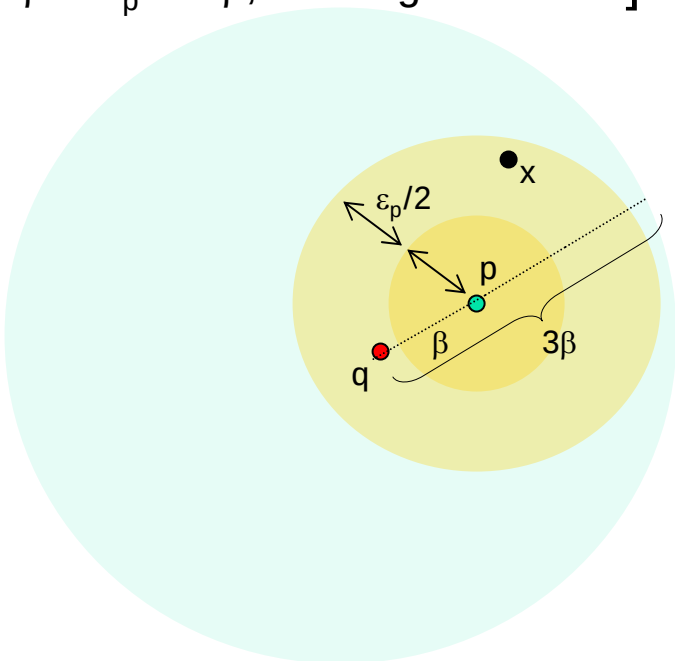
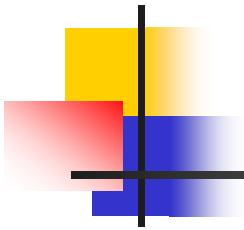
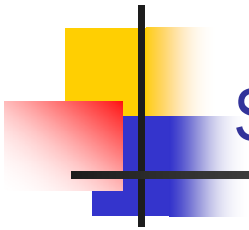


Figure: $x \in B(p,\varepsilon_p) \rightarrow x \in B(q,3\beta)$



Putting it all together

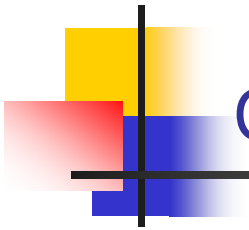
- Choose m = 10K log n, and K = 10(log n)4^d, then
  - We can confine the size of our search from n to n/10, so a recursive datastructure yields a depth of log n
  - Moreover, the failure probability p not close to q is at most $1/n^9$
  - => Query time of O(m log n) = O(4^d $\log^3$ n)



Summary

- There is extensive literature available which formalizes the concept of the intrinsic dimension of the data.

- We observe that if some nice conditions hold, we can cleverly construct fast algorithms for traditional problems such as NN.



Questions / Discussion

- All is well and good if we know the intrinsic dimension...

  but how do we know the dimension??

References

- [1] Clarkson, K. (2005) Nearest-Neighbor Searching and Metric Space Dimensions.

- [2] Cutler, C. (1993) A Review of the Theory and Estimation of Fractal Dimension. In H. Tong, editor, *Dimension Estimation and Models*. World Scientific.