IIT KHARAGPUR

IIT GANDHINAGAR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Scalable Data Science

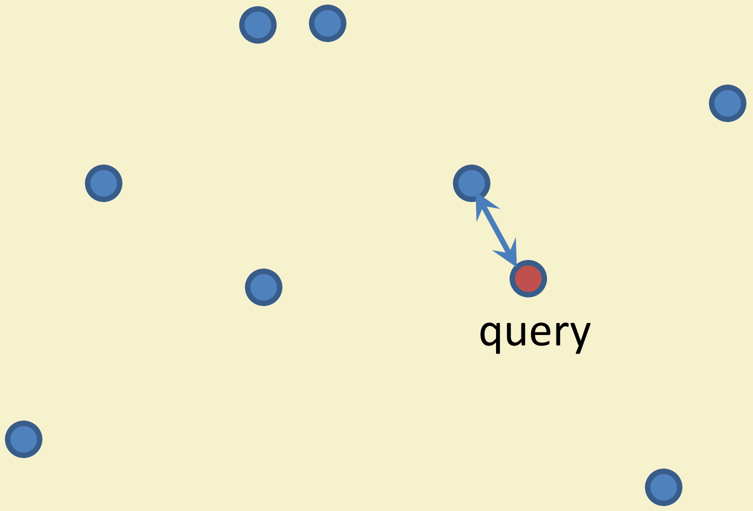
Lecture 11: Near Neighbors

Anirban Dasgupta

Computer Science and Engineering

IIT GANDHINAGAR

Finding Near Neighbors



Given a set of data points and a query

Can we find what is the nearest datapoint to the query?

- K-nearest neighbors
- d(p, query) < r

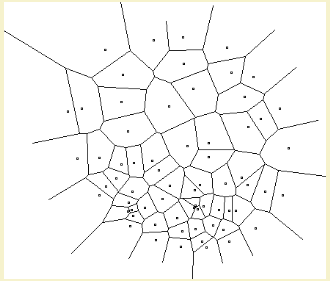
Applications

- Numerous
  - Finding near duplicate webpages / articles
  - Finding similar images in search
  - Clustering
  - Nearest neighbour classifier
- Variants
  - all pairs near neighbors

Naïve solution?

- Naïve scan
  - $O(nd)$ time for each query

- Can we calculate and store the Voronoi partition of the point-set?
  - Will give the exact answer if possible
  - needs $n^{d/2}$ storage for n points in d dimensions



IIT Gandhinagar
Indian Institute of Technology Gandhinagar
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

Space Partitioning trees

- Basic idea
  - Recursively partition the space
  - Given the query, prune the dataset using the created partition tree
  - All depends on how to partition

Kd-trees

- Works “well” for “low to medium” dimensions
- Initially proposed by Bentley 1970
- Originally, k was #dimensions
- Idea: each level of the tree uses a single dimension to partition

Algorithm

- Each level has a cutting dimension
- Cycle through the dimensions
- At every step, choose the point which is the median along that dimension, create an axis-aligned partition

Example

Complexity

- Space taken = $O(n)$
- Nearest neighbour search:
  - Defeatist search: only search the child that contain the query point
  - Descending search: maintain the current near neighbour and distance to it. Visit one or both children depending on whether there is intersection
  - Priority search: Maintain a priority queue of the regions depending on distance.
  - Can potentially take $O(n)$

Variants

- Several variants of space partitioning trees possible
  - Random Projection tree chooses a unit direction at random for every node
  - PD tree uses the principal eigenvector of the covariance matrix
  - 2-Mean tree : partition the data into 2 clusters, find the hyperplane that bisects the line connecting them

Possible intuition to analyze

- Does the partitioning algorithm adapt to “intrinsic dimension” ?
  - i.e. if the data has some low-dimensional structure
  - E.g. if the data has “intrinsic dimension” d, then all cells O(d) levels below a cell C has at most ½ the diameter of C

Possible intuition to analyze

- Does the partitioning algorithm adapt to "intrinsic dimension" ?
  - i.e. if the data has some low-dimensional structure
  - E.g. if the data has "intrinsic dimension" d, then all cells O(d) levels below a cell C has at most ½ the diameter of C
- Definition of "intrinsic dimension" is not obvious
  - Ex: covariance dimension is d if the d largest eigenvalues of covariance matrix account for $1 - \epsilon$ fraction of trace

Possible way to analyze

- Does the partitioning algorithm adapt to “intrinsic dimension” ?
  - i.e. if the data has some low-dimensional structure
  - E.g. if the data has “intrinsic dimension” d, then all cells O(d) levels below a cell C has at most ½ the diameter of C
- Definition of “intrinsic dimension” is not obvious
  - Ex: covariance dimension is d if the d largest eigenvalues of covariance matrix account for $1 - \epsilon$ fraction of trace
- Can be shown that RP, PD trees adapt to this dimension, but k-D tree does not

Summary

- Nearest neighbour question
- Number of algorithms for low dimensional data based on space partitioning trees
  - Some of the adapt to the intrinsic dimensionality of data

References:

- Primary references for this lecture
  - Foundations of multidimensional and metric data structures, H. Samet. Morgan Kaufman 2006.
  - “Which space partitioning trees adapt to Intrinsic Dimension”, Verma, Kpotfe, Dasgupta UAI 2009.

Thank You!!