YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CptSci 445/545: Introduction to Data Mining — *Handout #5*

V. Rokhlin — *November 11, 2013*

Problem Set 3

Due Date: December 9, 2013

Write a function (subroutine) that, given a collection of n points in the plane, finds the nearest k neighbors of each of the points. The assumption is that n is large (~ 1000000), and that examining all $n(n-1)/2$ pairs is not an option.

Start with constructing an adaptive quad-tree in the plane (linked list, tables, etc.). For each leaf in the tree, move up the tree, finding all the branches supporting it; then, move down the tree, ending at the original leaf, with the table of all leaves in the tree that might contain nearest neighbors. Finally, examine all points inside the "suspect" leaves, and find the k nearest neighbors for each of the points inside the original tree.

Include a short description of the algorithm in your report. In particular, briefly explain the data structure used to store your tree and how you use it when you search for neighbors. You may use diagram or a simple example. You do not have to discuss all the details of your implementation.

In FORTRAN, the calling sequence of your function should be

$$seek(a, n, k, iz), \tag{1}$$

where a is an $2 \times n$-matrix and iz is a $k \times n$-matrix.

In C, the calling sequence of your function should be

$$seek(double *\ a, int\ n, int\ k, int *\ iz), \tag{2}$$

where

a points to an array of doubles of size $2n$, containing $a(1,1), a(1,2)$,$a(2,1), a(2,2), \ldots, a(n,1), a(n,2)$, a being an $n \times 2$-matrix. For $i = 1, \ldots, n$, $a(i,1)$ is the first coordinate of point i, and $a(i,2)$ is the second coordinate of point i (INPUT PARAMETER).

n is the number of points (INPUT PARAMETER).

k is the number of requested nearest neighbors. k is an integer between 1 and $n-1$ (INPUT PARAMETER).

iz points to an array of integers of size kn, containing

$$iz(1,1), \ldots, iz(1,k), iz(2,1), \ldots, iz(2,k), \ldots, iz(n,1), \ldots, iz(n,k).$$

The memory is allocated by user. This is an OUTPUT PARAMETER.

REMARK 1. The k nearest neighbors of point i are the points $iz(i,1), \ldots, iz(i,k)$.

REMARK 2. No point is its own nearest neighbor.

In addition, your code should contain the function *seek_naive*, which has EXACTLY the same calling sequence as *seek*. *seek_naive* finds k nearest neighbors of each point by direct scanning ($O(n^2)$ algorithm). Note that given the same input, *seek* and *seek_naive* should return the same output.

The quad-tree you construct should have the following structure (as discussed in class):

- Each node in the tree is a box (a square in the plane).
- The root box contains all the n points.
- If a box contains more than k points, it should have 4 children. The child boxes result from dividing the parent box in half in each coordinate. For example, the children of $[0, 2] \times [0, 2]$ are the boxes $[0, 1] \times [0, 1]$, $[1, 2] \times [0, 1]$, $[0, 1] \times [1, 2]$ and $[1, 2] \times [1, 2]$.
- If a box contains between 0 and k points, it should have no children (a "leaf" box).