STAT 534 Homework 3
Due April 29, 2019
©Marina Meilă
mmp@stat.washington.edu

Reading:

Files: The data files `hw1-pb2.dat, hw1-pb3-x.dat,hw1-pb3-y.dat` are available on the web.

Problem 1 – Kernel density estimation

a. Write a program that:

1. reads a set of floating point numbers, the *observed points*, $x_1, x_2, \ldots x_n$, $n \leq 1000$ from file `hw3-pb1-x.dat` and stores them
2. reads a set of floating point numbers, the *query points*, $y_1, y_2, \ldots y_m$, $m \leq 1000$ from file `hw3-pb1-y.dat` and stores them
3. computes the kernel density estimate

$$f(y) \;=\; \frac{1}{n}\sum_{i=1}^{n} k\left(\frac{y - x_i}{h}\right)$$

for all $y_1, \ldots y_m$. In the above, k is the *Gaussian kernel*

$$k(z) \;=\; \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

and h is a parameter. For this program take $h = 0.1$.

The function $f(y)$ represents the kernel estimate of the density from which the observed points $x_1, x_2, \ldots x_n$ were sampled.

4. prints out all $f(y_i),\ i = 1, \ldots m$

The files have the same format as in problem 2.

b. Run the program and print the output.

c. How many kernel evaluations will the program perform for n observation points and m query points ?

Problem 2 – Radix trees

Consider a radix tree over the alphabet `a,b,c,...z` that stores at each node the following information: `p` a pointer to the parent node, `isword` a Boolean that is true when the node is the end of a word, a letter of the aalphabet representing the node label, and `children`, a sorted doubly linked list of tuples `(l, c)`, where `l` is a letter of the alphabet, and `c` is a pointer to the child node labeled with `l`. Hence, the same label is stored once by the parent and once by the child node.

The tree T maintains the following properties:

- the elements of `children` are sorted by the key `l`; there are no NULL pointers, i.e. the list has exactly one entry for each child of the current node.
- every node is part of some word in the dictionary (the dictionary being the set of words stored in T); in other words, for every leaf node `isword` is true.

a. Write in pseudocode the algorithm TREE-FIND$(T, a_1 \dots a_m)$ that returns a pointer x to the node corresponding to $a_1 \dots a_m$ if the node is in T and NULL otherwise.

b. Write in pseudocode the algorithm TREE-DELETE(T, x) that deletes the node x from T. You can assume that x was returned by a FIND call. Assume that the linked list `children` supports the following operations: IS-EMPTY (constant time), INSERT, DELETE, FIND (all $\mathcal{O}(l)$ time, where l is the list length). Ensure that this operation maintains the tree properties.

c. What is the worst case run time of TREE-FIND and TREE-DELETE on a tree containing n nodes?

d. Suppose you decided to store the label `l` only once, either at the parent or at the child. Which version would be the most efficient from the point of view of the TREE-DELETE algorithm?

e. Suppose T contains only words of length $\leq m$, $m > 1$, and that the alphabet contains $r = 26$ letters. Denote the *efficiency* of T the ratio w/n where n is the number of nodes and w is the number of words stored. What is the smallest efficiency attainable? What is the largest?