CMPE 365 Lab 3: Convex Hulls

Cache Angus

20000629

October 4, 2018

# Question 1:

When dealing with empirically arranged points, I found that the easiest solution to implement ended up being the Graham Scan method. To do the Graham scan, a random number of points, each with random x and y coordinates were generated and plotted. Python has a built-in plotting library so this was easily executed. After the generation of these points, and their subsequent insertion into a list, the points where then placed into a function cH that only received lists of points. This overall function housed most of the other functions utilized in finding the convex hull.

Following the steps of a Graham Scan Algorithm, a bottom point array was declared so that the lowest point among the set could be determined as the pivot point. The index of the point was found by iterating through the list of points among all x’s and y’s to replace the index every time a point with a lower value was found. After doing so, the points where then converted to their polar coordinates so they could be compared using their angle from the new origin at the bottom point. A function to find the distance between two points was also found to help the sorting process later.

The sorting process took in this new set of points and compared their angles to pivot angle. They were subsequently placed in one of three arrays, smaller, equal, or larger, and then recursively sorted until all were in the order to traverse by. If the angle was equal to the pivot, they were sorted by the distance from the pivot

One of the strategies we discussed in class was using the determinant of 3 points to find out if a third point is within the convex hull so the determinant was calculated between 3 points.

The input list of points was sorted into a new list and the bottom point was removed from said list. A starting convex hull was created with the bottom point and the first value in the sorted list. Iterating throughout the sorted list, while there was more than one value in the convex hull and the determinant of the latest value in the convex hull and the subsequent two was < = 0 (if the determinant is not positive, the middle point is creating a concave triangle), then the value is then taken off of the convex hull. As long as the entirety of the sorted list has not been run through, the next value of the sorted list is added to the convex hull.

The ratio of convex hull size to number of points, averaged over the different sets is decreasing as the number of values in a set is increasing. For one test with only 10 points, the ration was 0.8 while a second test with 30 points produced a value of 0.35.

# Question 2:

When repeating the previous question with Gaussian point distribution, it is clear to see that the ratio is relatively the same.

# Question 3:

To create the intersecting circles, first the center of each set of points was found by iterating through all of the points and finding the four cardinal points of the set. The essential radius of the points was also found by checking distances from the center. After these values were determined, a circle was created by finding the center, plotting the circle using the values obtained, and then patching it over the plot of the convex hull already there. Finding if there was an intersection or not came from comparing two circles created and finding the distance between their centers. If the distance between their centers was less than the combined value of their radii, then they intersected.

This was tested with random sets and some showed intersection.