Milestone 5

Ryan Stanley
Arif Topcu
Swana Weng

1. How will setting `MIN_DISTANCE = 0` affect the clusters? Try it and see! What is a reason to choose 0?

   For good clustering it is beneficial to have an excess of data points in areas where you spend more time, while having less data points in areas you are simply moving through. By not limiting the minimum distance, the phone collects data constantly on a chosen interval even if the user has been sitting in the same location for a couple of hours. This way you can distinguish between the trail of a person moving, which should have less data points, and an area where the person has spent more time.

2. Why do we include the `final` modifier for the list of points passed into the DBScan algorithm?

   The `final` modifier is used to convey that a given value should not be changed. When used with a method parameter, it becomes easier for a developer to read and understand what the method should be doing.

   Java methods are pass by value, which means that the value of the parameters passed into a method are copied and then run. This means that the values of the parameters will never change outside of the method. The `final` modifier adds to this limitation by not allowing the value of the parameter to be changed at all during the method, even though regardless it will still not be changed outside of the method (without other means).

   Interestingly, pass by value when used on object references only protects changing the reference itself, not what it is referencing. This is because it creates a copy of the reference passed in, which still references the real object. The use of the final modifier does not make a difference in this case either. For an example, take a look here: https://ideone.com/ZXE7SL (hopefully the link stays active long enough!).

3. The DBScan code infers generic type `T` so that the clustering algorithm can work on other data points, not only GPS coordinates. But under what condition will it work? In other words, what must `T` define in order for the list to be a valid parameter for `DBScan`?

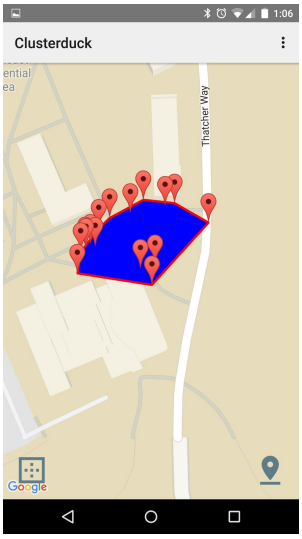
   `T` must implement `Clusterable<T>` in order to be a valid parameter for `DBScan`. This means that `T` must have a `double distance(T other)` method implemented.

4. Why do we use a `HashMap` to store the state of each point in the DBScan algorithm?

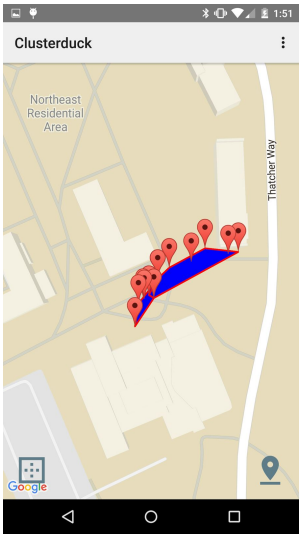
   The DBScan algorithm requires multiple accesses to the state of a given point. As `HashMap`s are touted as being perfect for quick accesses when given a key, they are the most logical implementation.

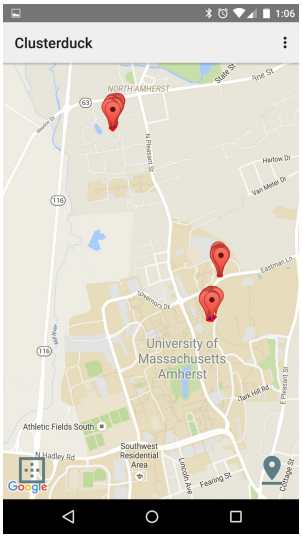
5. Collect a bunch of points that are relatively spread out, for instance some around the CS building and some at your home. In this case you'd expect to see two clusters: one at the CS building and one at home.
   a. Visualize the map and submit a screenshot of the clusters for at least two choices of `epsilon` and `minPts` parameters. For most Android devices, holding the power button and home button simultaneously will take a screenshot.
   b. Briefly describe how each parameter changes the resulting clusters.

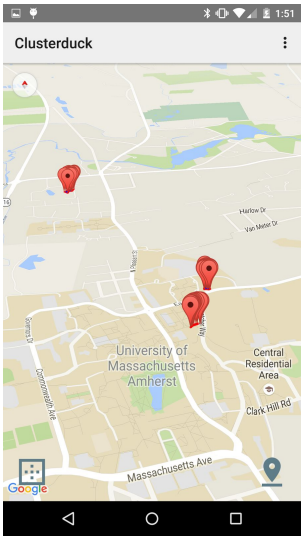
eps 50, minPts 15



eps 30, minPts 15







eps, when lowered, decreases the size of any given point's neighborhood. This can be useful to prevent counting points retrieved while moving, however it can limit possible cluster areas as seen in this example.

minPts decreases the minimum points required for a neighborhood to become (or join) a cluster. In this case we did not change the minPts value and focused on only visualizing the changes of the eps value, however if we were to decrease minPts we may see areas become clusters that were once considered noise. An example could be our car stopped at a stop sign or a red light.

6. What is one other possible application of DBScan you could imagine?

   DBScan is useful for data types that have a definable distance between each other. In this application a literal distance was used, however it is possible to use subjective distances with other subjects. For example, it is possible to define the distance between two given words by the similarity of their meaning and usage. This example could be used to create word clouds or suggest alternatives to words while writing. Another example could be an implementation of the six degrees of separation (or even the six degrees of Kevin Bacon), where the distance between two people is defined by how direct their relationship is. Two people that are friends would have a distance of 1, while two people that are only connected through two other people in a chain would have a distance of 3.

7. Describe how the workload was distributed among the team members.

Reading Documentation/Leading Discussion:	Swana
Lab Rat:	Arif
Data Analytics:	All
Algorithm/Implementation Design:	Arif and Ryan
Peer Programming:	All

Other notes:

While we did not spend time on any of the suggested extra credit options, we did try our best to make a clean user interface and codebase. Upon load, the application will zoom into the last known location of the user. The user is presented with three buttons: a ‘cluster’ button on the left side of the screen that re-clusters the data points, a ‘track location’ button that toggles data collection, and a settings menu with the option to clear all stored locations from the database. On the code end, we added a runnable that will run once every second and attempt to connect to `ContextService` until it successfully connects. We found this favorable to trying to connect again when the user requests a service as this application attempts to get the user’s last known location upon startup. Unfortunately we cannot simply try to connect to `ContextService` on `MapsActivity`’s `onCreate` or `onResume`, as `ContextService` will not have started.

To try out our application without going through and changing the Google Maps API key, we included two signed copies. We have not had 100% success in running the release copy on other android phones, however debug seems to work just fine. For an added treat, try reclustering with your sound turned on!