Machine Learning 381 Fall 2016 10/4/16

Qiman Wang

K Means Clustering Algorithm

The outline of the K-means Clustering algorithm pseudocode is as follow.

1. Read in the data file, store each data as an object. The component of each data set corresponds to the variable of each object.
2. Store each data point into an array.
3. We Assign random K value (1,2,3) to each data point. This will be used to compute the starting centroid.
4. compute the centroid of all data point with the same K value. The centroid is a made up data point with same number of components as the data points (7 in this case). The component of the made up data point is obtained by the average of every data point’s component that has the same K value.
5. Once the first starting random centroid is obtained, we can calculate IV and EV. IV is obtained by adding up 3 numbers. The 3 numbers are generated separately using the 3 centroids. For each centroid in each corresponding K group, we sum the Euclidean distance of every data point in that same K group from centroid. Once we do get the sum from each group, we add it up to obtain the total IV. It makes sense because we want to decrease IV, meaning to decrease the distance between all the data points and their centroid. EV is calculated by summing the distance of every data point that is not in the same K group. Since we use a for loop to compare every point, we will get a duplicate distance .So therefore when we divide by the total data point (210 samples) we divide by 420 instead to remove the duplicate.

IV/EV is the ratio and our job is to minimize IV, or increase EV. Which will lower the ratio and that’s the objective.

1. Once we have the first centroid, we will now keep repeating the steps of updating each data Point’s cluster pile number. And then calculate the new centroid because the data points in the same cluster piles have changed. Every time these 2 process are repeated, we print out the iteration number and IV, EV, IV/EV values. We repeat this step until no data point’s moved to another cluster. Hence we use while loop.
2. The program will terminate when No data points are moving to another cluster pile. Which means the centroids for each of the K cluster piles have converged to a data point. Also IV have been minimized, EV have been maximized and IV/EV have been minimized.

After running this program many times, the centroids converge in about 5-10 iterations. So on average 10 iterations. IV converge around 313, EV around 389 and IV/EV around .80. These results are obtained by assigning each data point to one of the 3 cluster piles. Then generate the 3 starting centroids. Below is a sample Program run.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteration | area | perimeter | compact | length | width | asymmetry | Length  groove | IV | EV | IV/EV |
| 1 |  |  |  |  |  |  |  | 683.41 | 306.52 | 2.2296 |
| Centroid 1 | 14.6429 | 14.4748 | 0.8684 | 5.9655 | 3.2249 | 3.6900 | 5.4017 |  |  |  |
| Centroid 2 | 14.6564 | 14.4775 | 0.8692 | 5.6103 | 3.2348 | 3.5031 | 5.3788 |  |  |  |
| Centroid 3 | 15.1546 | 14.6886 | 0.8742 | 5.6660 | 3.0302 | 3.8846 | 5.4386 |  |  |  |
| 2 |  |  |  |  |  |  |  | 339.21 | 382.68 | 0.8864 |
| Centroid 1 | 12.0797 | 13.3413 | 0.8513 | 5.2583 | 2.8857 | 4.9315 | 5.1038 |  |  |  |
| Centroid 2 | 13.8976 | 15.0910 | 0.8778 | 5.4363 | 3.1857 | 2.2386 | 5.0467 |  |  |  |
| Centroid 3 | 18.0273 | 15.9949 | 0.8838 | 6.1011 | 3.6470 | 3.6517 | 5.9439 |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |
| Centroid 1 | 11.9156 | 13.2527 | 0.5215 | 5.2237 | 2.8668 | 4.9461 | 5.0848 | 319.20 | 389.41 | 0.8197 |
| Centroid 2 | 114.2152 | 15.2596 | 0.8772 | 5.4971 | 2.2188 | 2.4978 | 5.1194 |  |  |  |
| Centroid 3 | 18.3537 | 16.1397 | 0.8842 | 6.1554 | 306836 | 3.6404 | 6.0073 |  |  |  |
| 4 |  |  |  |  |  |  |  | 314.93 | 390.32 | 0.8068 |
| Centroid 1 | 11.9186 | 13.2568 | 0.5122 | 5.2256 | 2.8632 | 4.8855 | 5.0875 |  |  |  |
| Centroid 2 | 14.3918 | 14.3364 | 0.8786 | 5.5243 | 3.2455 | 2.5957 | 5.1451 |  |  |  |
| Centroid 3 | 18.5398 | 16.2233 | 0.8842 | 6.1817 | 3.7019 | 3.6120 | 6.0447 |  |  |  |
| 5 |  |  |  |  |  |  |  | 314.28 | 390.40 | 0.8050 |
| Centroid 1 | 11.9376 | 13.5685 | 0.8511 | 53231 | 2.8661 | 4.839 | 5.0960 |  |  |  |
| Centroid 2 | 14.4676 | 14.3645 | 0.8796 | 5.5322 | 3.2586 | 2.5960 | 5.1488 |  |  |  |
| Centroid 3 | 18.5754 | 16.2394 | 0.8842 | 6.1865 | 3.7055 | 3.6097 | 6.0465 |  |  |  |
| 6 |  |  |  |  |  |  |  | 313.52 | 390.32 | 0.8032 |
| Centroid 1 | 11.9397 | 13.2669 | 0.8515 | 5.2292 | 2.8671 | 4.8040 | 5.0548 |  |  |  |
| Centroid 2 | 14.5764 | 14.4108 | 0.8792 | 5.5457 | 3.2667 | 2.6482 | 5.1668 |  |  |  |
| Centroid 3 | 18.6519 | 16.2675 | 0.8850 | 6.1987 | 3.7154 | 3.5885 | 6.0560 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  | 313.53 | 390.23 | 0.8035 |
| Centroid 1 | 11.9397 | 13.2669 | 0.8515 | 5.2292 | 2.8671 | 4.8040 | 5.0955 |  |  |  |
| Centroid 2 | 14.5762 | 14.4252 | 0.8791 | 5.5526 | 3.2694 | 2.6398 | 5.1765 |  |  |  |
| Centroid 3 | 18.6845 | 16.2804 | 0.8851 | 6.2010 | 3.7148 | 3.6135 | 6.0589 |  |  |  |
| 8 |  |  |  |  |  |  |  | 313.22 | 390.21 | 0.8028 |
| Centroid 1 | 11.9644 | 13.2748 | 0.8522 | 5.2293 | 2.8729 | 4.7297 | 5.0885 |  |  |  |
| Centroid 2 | 14.6485 | 14.4604 | 0.8792 | 5.5638 | 3.2779 | 2.6489 | 5.1923 |  |  |  |
| Centroid 3 | 18.7218 | 16.2974 | 0.8851 | 6.2089 | 3.7227 | 3.6036 | 6.0661 |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

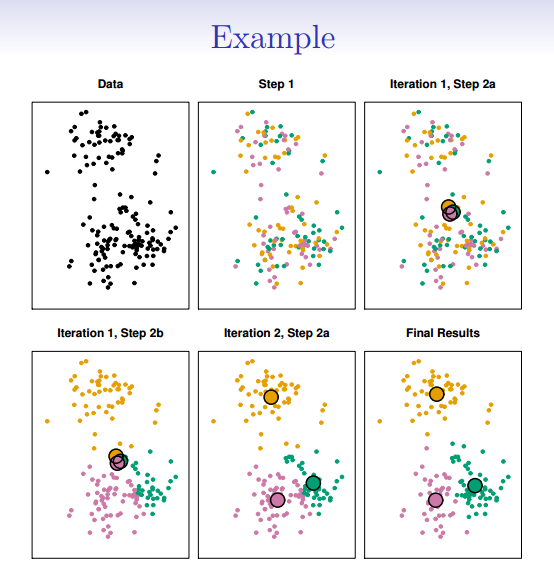
This data set values converged t around 5th iteration. The next 3 iteration didn’t change the values too much, but there was still a few data points that are in the wrong cluster pile. Once the centroids converge, the IV/EV have also been minimize and all the data points are in the correct cluster pile. The first 3 centroids are around the same and IV/EV is high.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  | 683.41 | 306.52 | 2.2296 |
| Centroid 1 | 14.6429 | 14.4748 | 0.8684 | 5.9655 | 3.2249 | 3.6900 | 5.4017 |  |  |  |
| Centroid 2 | 14.6564 | 14.4775 | 0.8692 | 5.6103 | 3.2348 | 3.5031 | 5.3788 |  |  |  |
| Centroid 3 | 15.1546 | 14.6886 | 0.8742 | 5.6660 | 3.0302 | 3.8846 | 5.4386 |  |  |  |

After one iteration, the centroids moves very closely to the correct centroid point .

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 |  |  |  |  |  |  |  | 339.21 | 382.68 | 0.8864 |
| Centroid 1 | 12.0797 | 13.3413 | 0.8513 | 5.2583 | 2.8857 | 4.9315 | 5.1038 |  |  |  |
| Centroid 2 | 13.8976 | 15.0910 | 0.8778 | 5.4363 | 3.1857 | 2.2386 | 5.0467 |  |  |  |
| Centroid 3 | 18.0273 | 15.9949 | 0.8838 | 6.1011 | 3.6470 | 3.6517 | 5.9439 |  |  |  |

So for the manual picking the starting centroid, I picked centroid similar to the centroid in iteration 2. This helped the centroids converge even faster than randomly choosing the starting centroids. The starting centroid close to iteration 2 gave the best result because all the data points are around one of the centroid points in iteration 2. Some data fall between centroid 1 and 2. Some between centroid 2 and 3. There will also be a few data point lower than centroid 1 and higher than centroid 3. The distance can be treated as the length and the length is used to get the area. All data point that is closest to a centroid will be in that area. If the points are in the right cluster area since the beginning, then we only have to move a few wrong data points to the right cluster pile aka to the correct area location.



From the above picture we can visualize the steps better. We can see that the iteration step can be a lot after if we choose the centroids apart from each other. Another condition is that the three centroids will be similar to the low, mid, and high data points. Another observation is that we can increase K, meaning more cluster pile and centroids. But at some point it will not be beneficial and if we plot the number of cluster and the cost function. At some k we will get a very insignificant increase in usefulness. And the graph look like a elbow, hence called elbow method. It is at the elbow joints, that we should pick the value K.