Imbalanced K Means Algorithm

Zou Xueyan 13253956

June 12, 2016

Abstract

First, the report would illustrate why changing the cluster density itself will not influence the error. Second, as in the previous report, we have observed a "jump" in error when increasing the cluster standard deviation. However, I have found that this "jump" is subject to the distribution of data points, so that it is very difficult to estimate when the jump would occur. ...

P.S. The parameter demonstration part is unchanged according to the previous report.

1 Introduction

In this section, parameters used in the report would be demonstrated, then comes the brief introduction of the experiment procedure.

1.1 Parameter Demonstration

In Figure 1, $(x_1, y_1) = (3, 3)$, which is set manually. Then base on $(x_1, y_1) = (3, 3)$, $(x_2, y_2) = (x_1 + d, y_1)$, where d is the distance between two cluster centers.

n_1, n_2 is the number of data in $cluster_1$ and $cluster_2$ respectively. For example, point (x_i, y_i) is in $cluster_1$, then $(x_i, y_i) = (x_1 + x_t, y_1 + y_t)$, where $t \in [0, 50)$, and x_t is generated randomly with mean=0, standard deviation = σ_1(σ_1 and σ_2 are the standard deviation of $cluster_1$ and $cluster_2$) in normal distribution.

Then we define $d = 4\sigma_1 + 4\sigma_2$, which could offer more than 99.9% confidence that two clusters will not overlap.

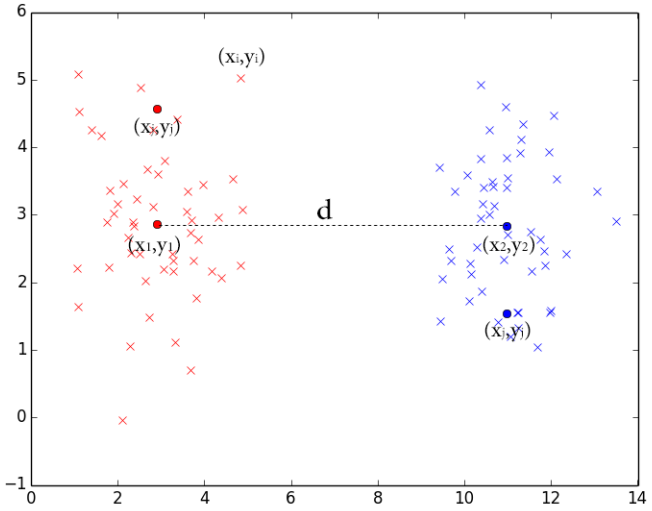

Figure 1: Initial cluster

(x_j, y_j) is the initial seed point of Kmeans algorithm. In order to exclude the influence of the initial position of seed points, $(x_j, y_j) = (x_1, y_1 + 2/3 * max(y_t))$ so that the seed points are ensured to appear in the clusters.

The cluster density ρ is defined as the number of points inside unit area.

$$\rho_1 = \frac{n_1}{\pi\sigma_1^2} \tag{1}$$

For the measurement of error ε, we define it to be the distance between the real center of two clusters and the final position of the seed points. The cluster density ρ is defined as the number of points inside unit area.

$$\sqrt{(x_1 - x_j)^2 + (y_1 - y_j)^2} \tag{2}$$

1.2 Experiment Introduction

The first experiment estimate the relationship between error and and the cluster density making the cluster standard deviation(scale) unchanged and without cluster overlap. The result shows that, unless the cluster overlap, cluster density imbalance would not influence the error.

The second experiment mainly focus on when there would exist a jump, and base on the result I would give an reasonable illustration why this situation would occur, and why this situation is unsolvable in the real world.

2 Experiment: Error and cluster density

In this section, with the standard deviation and distance between clusters $d = 4\sigma_1 + 4\sigma_2$ unchanged, the density of $cluster_2$ increased by 1 each time. Density ratio means the number of points in $cluster_1$ divided by the number of points in $cluster_2$.

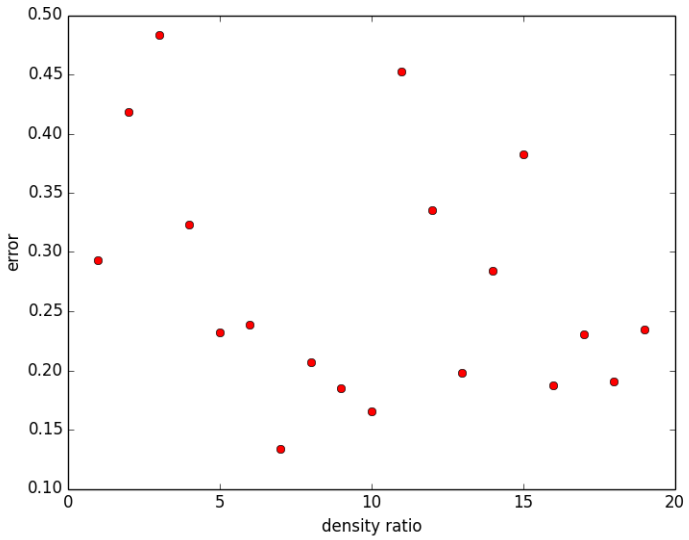

Figure 2: density ratio and error

As we can observe in the graph, as the density ratio increase, error diverse from 0.1 to 0.5 without special pattern, which means that neither there is a

relation between error and density ratio, nor would the density ratio influence the performance of the algorithm. The reason for this situation to happen is as the following. Kmeans Algorithm will distribute the nearest centroid to one point, so that as shown in the figure all the points to the left of the midperpendicular line would be assigned to the red centroid, and all the points to the right of midperpendicular line would be allocated to the blue centroid. So that, as long as two clusters doesn't overlap, the performance of the algorithm will not be influenced.

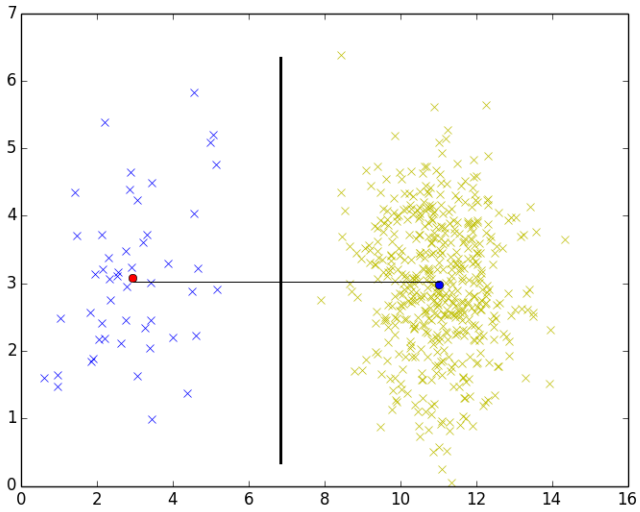

Figure 3: imbalanced clusters

3 Experiment: Explain the "jump"

As we observed in the previous experiment, when we increase the standard deviation, there would be a "jump" of error rate, as shown in figure 4.

The reason for this jump is that before that point, Kmeans algorithm works well, locating the training center to the actual center, but after the "jump" point, the error rate start to increase linearly, as shown in the data, $error = distance$, which linearly increased with standard deviation. The

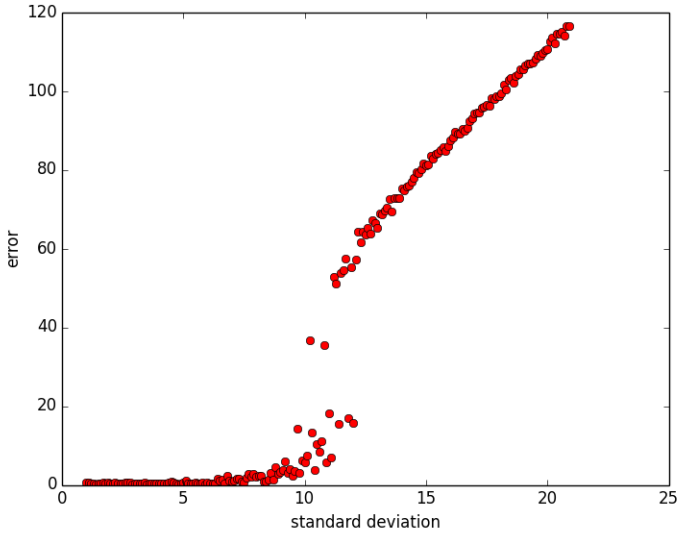

Figure 4: standard deviation and error(jump)

jump point situation is like the following:

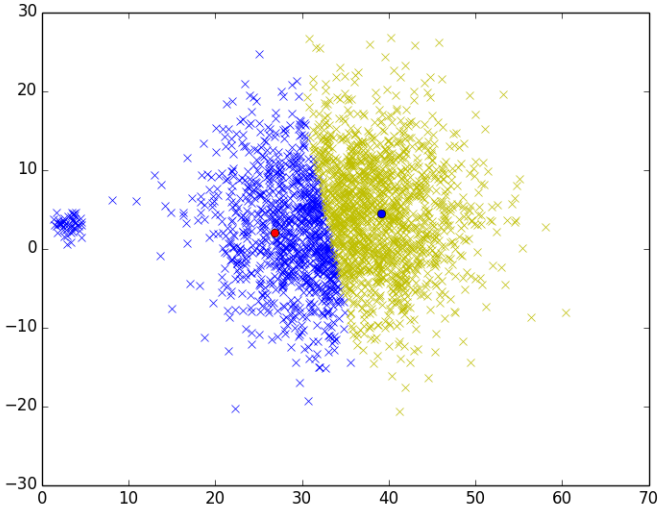

Figure 5: clustering result of imbalanced clusters

Actually this is a either or situation, either the cluster distribute well like figure 3, or wrongly distributed like figure 5. Then what is the driven force for Kmeans algorithm to perform poorly. The stop condition for Kmeans algorithm is either it has iterated a certain number of points, or the e has become a small number, which means that the centroids nearly unchanged.

If we assume two clusters don't overlap, unless in the first iteration, there are points in $cluster_2$ distributed to $cluster_1$, then the centroid of $cluster_1$ will driven away from its actual points, then in the next iteration new points are wrongly included into $cluster_1$, finally, it will be end up in the situation like figure 5.

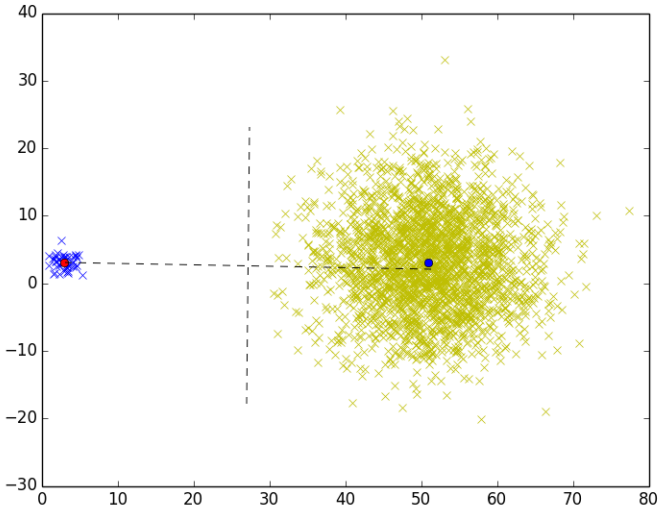

Figure 6: imbalance cluster

If we redefine $distance = n(\sigma_1 + \sigma_2)$, as we have mentioned data points would belong to the centroid where midperpendicular line take it apart. If the following situation like figure 6 happens, Kmeans algorithm would perform well, because all the points in $cluster_2$ falls on the right hand side of the midperpendicular line. If:

$$d/2 > 3\sigma_2 \tag{3}$$

This situation would always happen no matter what the standard deviation is. The experiment results also proves this statement. As the cluster centroid in $cluster_1$ is gradually moved to the right clusters, so it is very import to measure whether there is wrong clustering in the first round. Below is the graph of number of wrong estimate in the first round, and error rate, which shows that if there is 3-4 wrong estimation in the first round, it is very likely that the algorithm doesn't work. But exactly how many point would make a wrong prediction is base on the distribution of the clusters. This problem is really difficult to estimate in the real world, so the question when there is a jump is very difficult to predict.

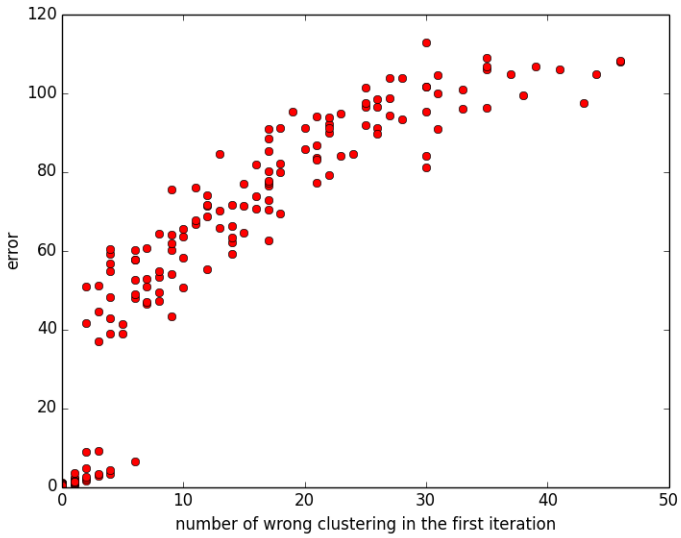

Figure 7: first round wrong estimate and error