Weekly Journal

Leila Uy

June 21, 2021

1 Work Update

This journal is going to be short because I have focused a lot of my time on the Literature Review.

1.1 Code Base

I spent some time on the code base for a serial k-means clustering algorithm in an R notebook. Since our last meeting on Friday, I have not touched the code because I wanted to finish my Literature Review first. Jishnu gave me an interesting article which provided pseudocode that will be useful when we implement our algorithm.

Figure 1: Pseudocode for a serial k-means clustering from [1]

Result: $C = \{C_1, \ldots, C_k\}, \quad c_j, j \in 1, \ldots, k$
c_j = random x_i *in* χ, $\quad j = 1, \ldots k, \quad c_j \neq c_i \quad \forall i \neq j$;
do
  $C_j = \emptyset, \quad j = 1, \ldots, k$;
  foreach $x_i \in \chi$ **do**
    $j = argminD(c_j, x_i)$;
    $C_j = C_j \cup x_i$
  end
  foreach $c_i \in C$ **do**
    $c_i = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$;
  end
while *convergence*;

Figure 2: Pseudocode for a parallel k-means clustering from [1]

Result: $C = \{C_1, \ldots, C_k\}, \quad c_j, j \in 1, \ldots, k$
if $threadID = 0$ **then**
  c_j = random x_i *in* χ, $\quad j = 1, \ldots k, \quad c_j \neq c_i \quad \forall i \neq j$;
end
synchronize threads;
do
  foreach $x_i \in \chi_{threadID}$ **do**
    $l_i = argminD(c_j, x_i)$;
  end
  synchronize threads;
  if $threadId = 0$ **then**
    foreach $x_i \in \chi$ **do**
      $c_{l_i} = c_{l_i} + x_i$;
      $m_{l_i} = m_{l_i} + 1$;
    end
    foreach $c_j \in C$ **do**
      $c_j = \frac{1}{m_j} c_j$;
    end
    if *convergence* **then**
      signal threads to terminate;
    end
  end
while *convergence*;

Currently the code uses for loops in R which I want to change to apply because it makes for more readable code and since apply does the for loops in C, it is slightly faster. I am also going to terminate my single processor instance on EC2 because there is constant charges for the EBS storage even when the instance is stopped and I am going to move up to a 4 processor instance (m4.xlarge).

For the assignment of each row of the data frame (pixel of the raster) to a cluster, I am thinking of using a format of dict{clusternumber: [list of pixels]} because it is easier to visualize with n number of

variables. I think the important thing while implementing the clustering algorithm is to make sure that we allow for the flexibility of n number of variables and k number of clusters.

2 Literature Review

Rather than looking for new articles for my Literature Review, I looked at the citations of my old articles to expand upon certain points. For example, an old article I had from a previous week [3] talked about k-means clustering in great lengths and even went on to explain the different extensions of k-means clustering. I know from the article the summary of the different extensions and what makes them unique like fuzzy c and kmeans++ but I wanted to go back to find the articles and skim through the papers. This means that I am able to cite them if I need to and I know if the algorithms are suitable for us to implement in the future [2, 4, 5].

I am finished my Literature Review and Diljot, Jishnu, Valerie and I are sending our Literature Reviews to each other for editing on Tuesday.

References

[1] Salvatore Cuomo, Vincenzo De Angelis, Gennaro Farina, Livia Marcellino, and Gerardo Toraldo. A gpu-accelerated parallel k-means algorithm. *Computers & electrical engineering*, 75:262–274, 2019.

[2] James Churchill Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of cybernetics*, 3(3):32–57, 1973.

[3] Anil Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[4] Dan Pelleg and Andrew Moore. Accelerating exact k -means algorithms with geometric reasoning. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 277–281. Association for Computing Machinery, 1999.

[5] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.