Lab 3 - Parallelizing k-means Stat 215A, Fall 2017

Melody Liu

October 23, 2018

- File list: Lab3.R: R file with similarity function that returns the Jaccard coefficient of two binary matrices and the file also contains calculation of similarity score with 100 iterations and maximum number of k equals to 10.

  Similarity_output_R.csv: I stored the result after running lab3.R file. The result is a 100 times 9 data frame, where row indicates all iterations and column indicates the number of clusters.

  Lab3_C.cpp: CPP file with similarity function.

  lab3_C.R: I sourced CPP similarity function and the file also contains calculation of simialrity score with 100 iterations and maximum number of k equals to 10.

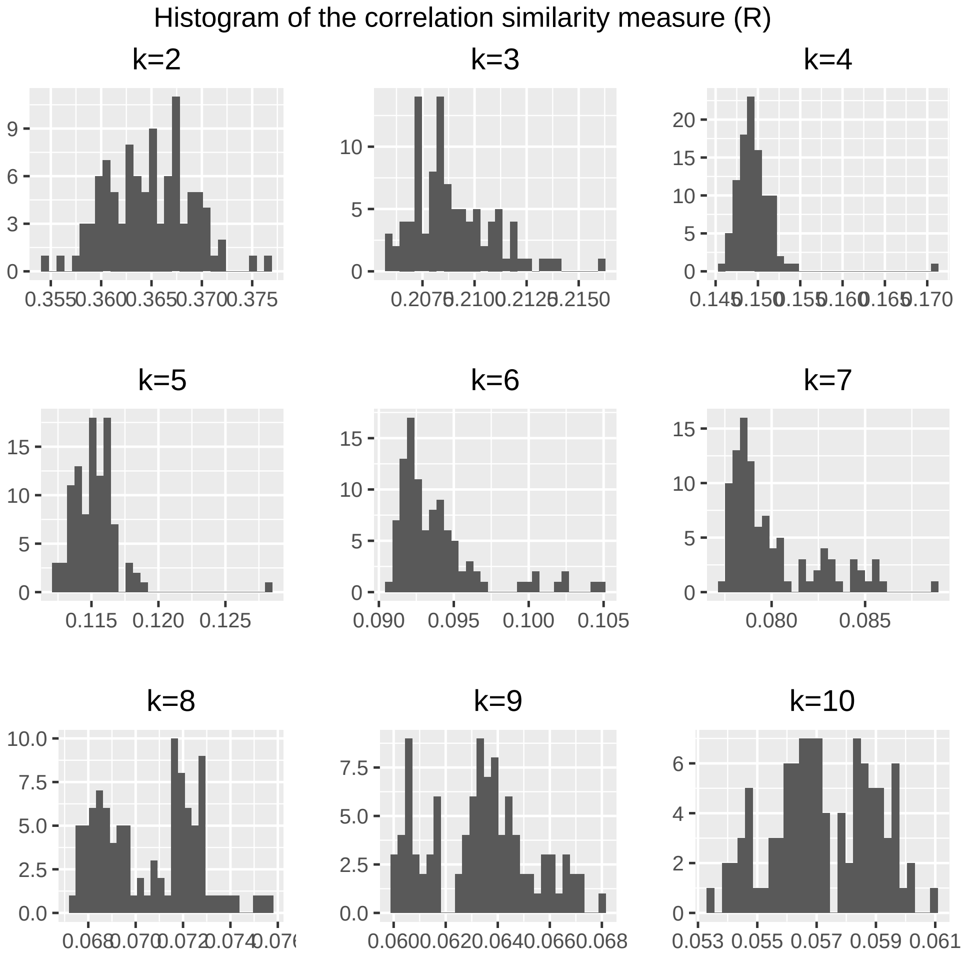
  Similarity_output_C.csv: I stored the result after running lab3_C.R file. The result is a 100 times 9 data frame, where row indicates all iterations and column indicates the number of clusters.

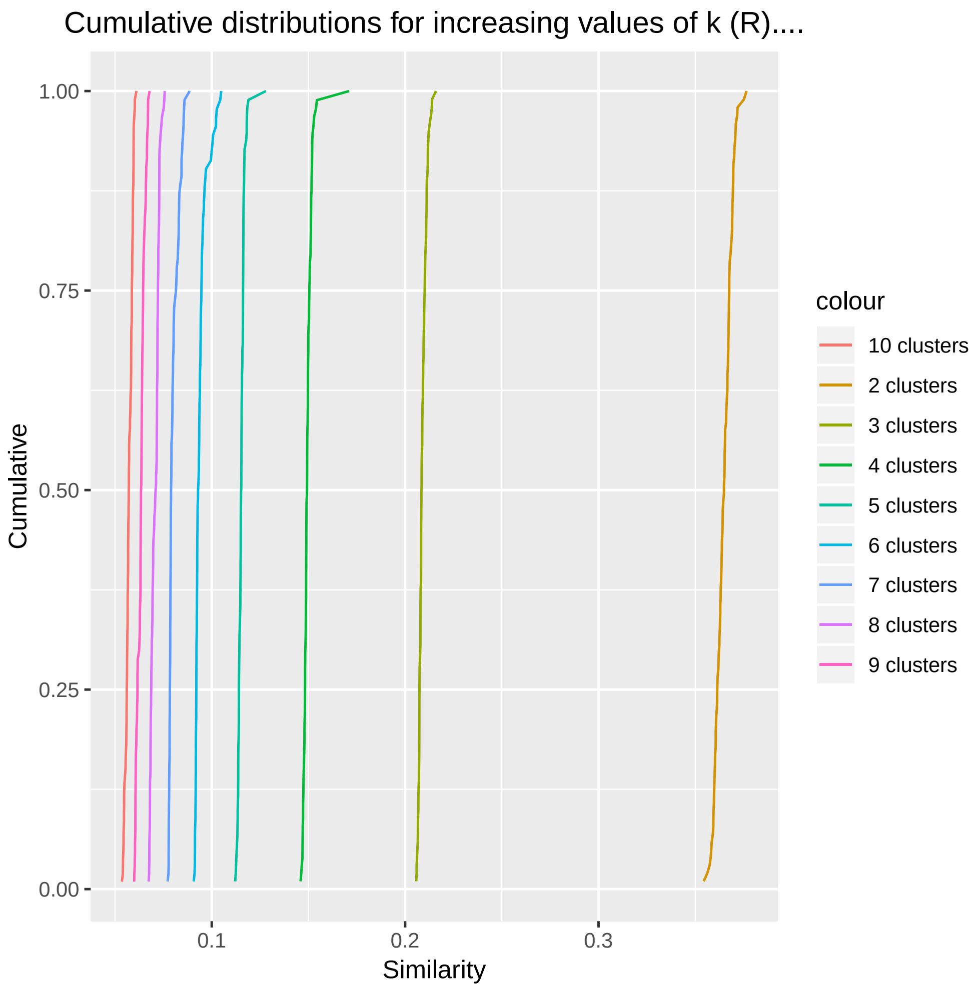
- Compare CPP. and R versions of the similarity matrix Due to the limit of nodes available on server, I subsetted first 10,000 rows of data from lingBianry data set and ran the code on my local machine. It took me around 23 mins to finish running the R code. The similarity score output is saved into a csv file. It took me 17 mins to finish running the code with similarity function in CPP. In CPP, I used a nested for loop to calculate N01, N11 and N10 instead of storing matrices. CPP is a compiled language while R is an interpreted language thus it is much faster for CPP. to run loops than R. The two methods yielded similar results: the similarity matrix generated from two methods are in line with each other.

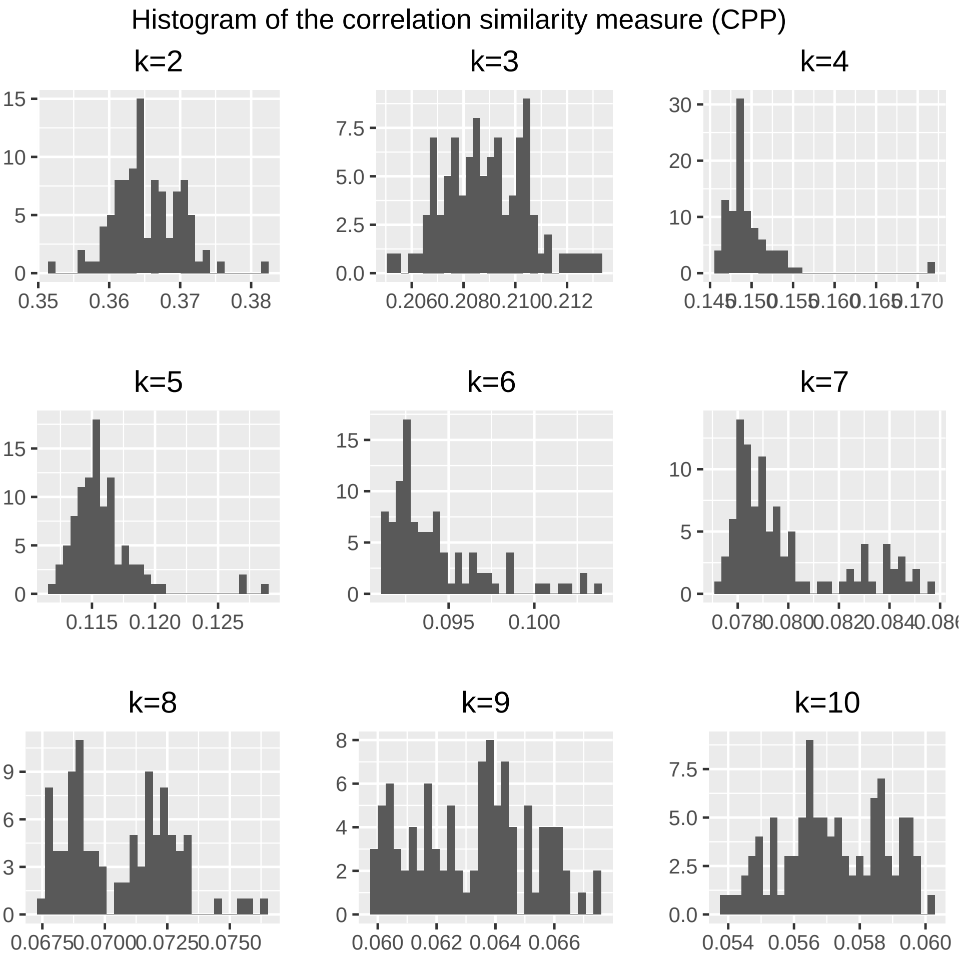
  I also ran the full data set with similarity function in CPP. It took me around 7 hours in my local machine. Results and plots will be shared in the report.

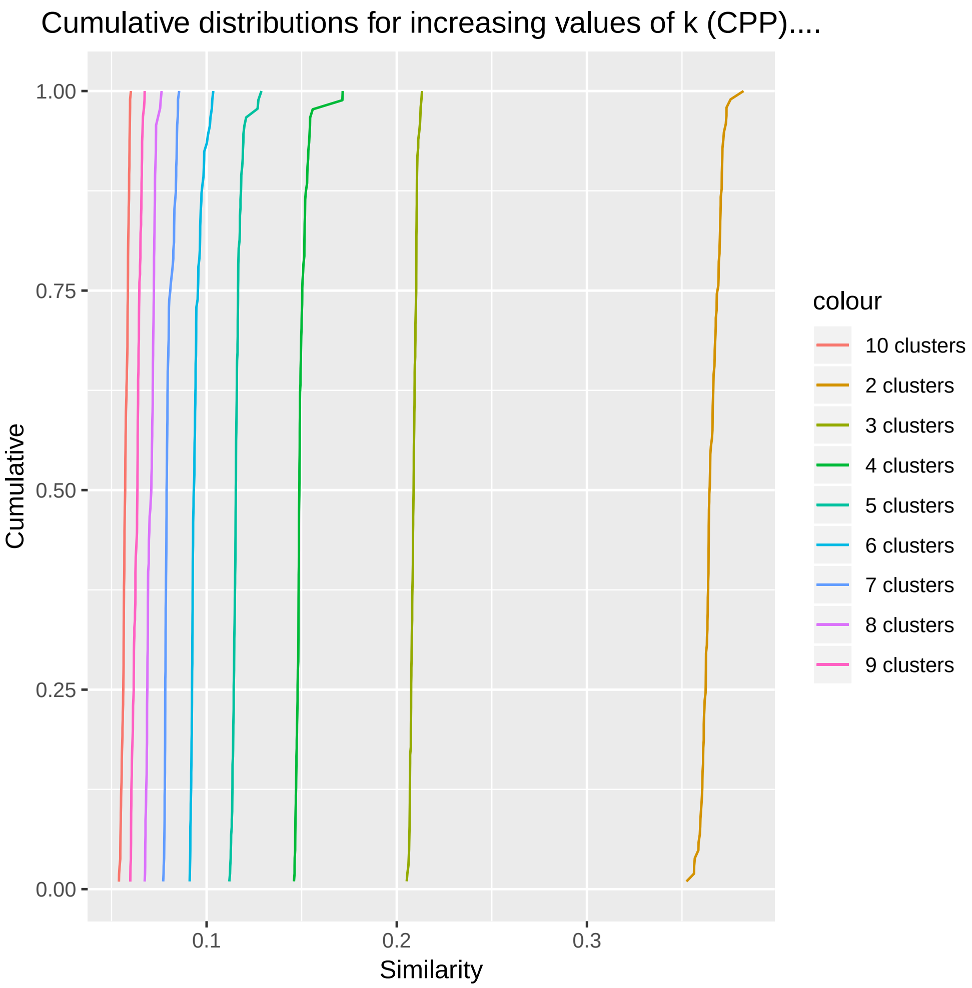
- make the plot for part 3

  The first two plots are for similarity function in R and last two plots are for similarity function in CPP.


Histogram of the correlation similarity measure (R)
k=2
0
3
6
9
0.355
0.360
0.365
0.370
0.375
k=3
0
5
10
0.2075
0.2100
0.2125
0.2150
k=4
0
5
10
15
20
0.145
0.150
0.155
0.160
0.165
0.170
k=5
0
5
10
15
0.115
0.120
0.125
k=6
0
5
10
15
0.090
0.095
0.100
0.105
k=7
0
5
10
15
0.080
0.085
k=8
0.0
2.5
5.0
7.5
10.0
0.068
0.070
0.072
0.074
0.076
k=9
0.0
2.5
5.0
7.5
0.060
0.062
0.064
0.066
0.068
k=10
0
2
4
6
0.053
0.055
0.057
0.059
0.061


Cumulative distributions for increasing values of k (R)....
Cumulative
1.00
0.75
0.50
0.25
0.00
0.1
0.2
0.3
Similarity
colour
10 clusters
2 clusters
3 clusters
4 clusters
5 clusters
6 clusters
7 clusters
8 clusters
9 clusters


Histogram of the correlation similarity measure (CPP)
k=2
15
10
5
0
0.35
0.36
0.37
0.38
k=3
7.5
5.0
2.5
0.0
0.2060.2080.2100.212
k=4
30
20
10
0
0.145.150.155.160.165.170
k=5
15
10
5
0
0.115
0.120
0.125
k=6
15
10
5
0
0.095
0.100
k=7
10
5
0
0.0780.0800.0820.0840.08
k=8
9
6
3
0
0.06750.07000.07250.0750
k=9
8
6
4
2
0
0.060
0.062
0.064
0.066
k=10
7.5
5.0
2.5
0.0
0.054
0.056
0.058
0.060



- discuss how many clusters to choose

  From the cumulative plot, we can conclude that similarity score is higher and more robust when k is smaller. The highest similarity score is achieved when k=2. When k=2, similarity scores mostly fall into the range of 0.35 to 0.375, which is relatively the highest among all k values. Thus k=2 is the optimal value for this data set.

- discuss whether you trust the method or not

  I trust this clustering method, as my last lab report showed 3 noticable cluster centers. Thus k=2 falls into the reasonable range. I also adopted both R and CPP functions to generate similarity coefficients and both gave me similar results.