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

SID: 24092167

October 23, 2017

1 Comparing C++ and R Versions of the Similarity Matrix

Before we proceed to tackle the first part of this lab, let us first take a brief detour to the second part of this lab, comparing the C++ and R versions of the similarity measure introduced by Fowlkes and Mallows. This similarity matrix takes the form:

$$cor(L_1, L_2) = \frac{\langle L_1, L_2 \rangle}{\sqrt{\langle L_1, L_1 \rangle \langle L_2, L_2 \rangle}}$$

Given that:

$$\langle L_1, L_2 \rangle = \langle C^{(1)}, C^{(2)} \rangle = \sum_{i,j} C_{ij}^{(1)}, C_{ij}^{(2)}$$

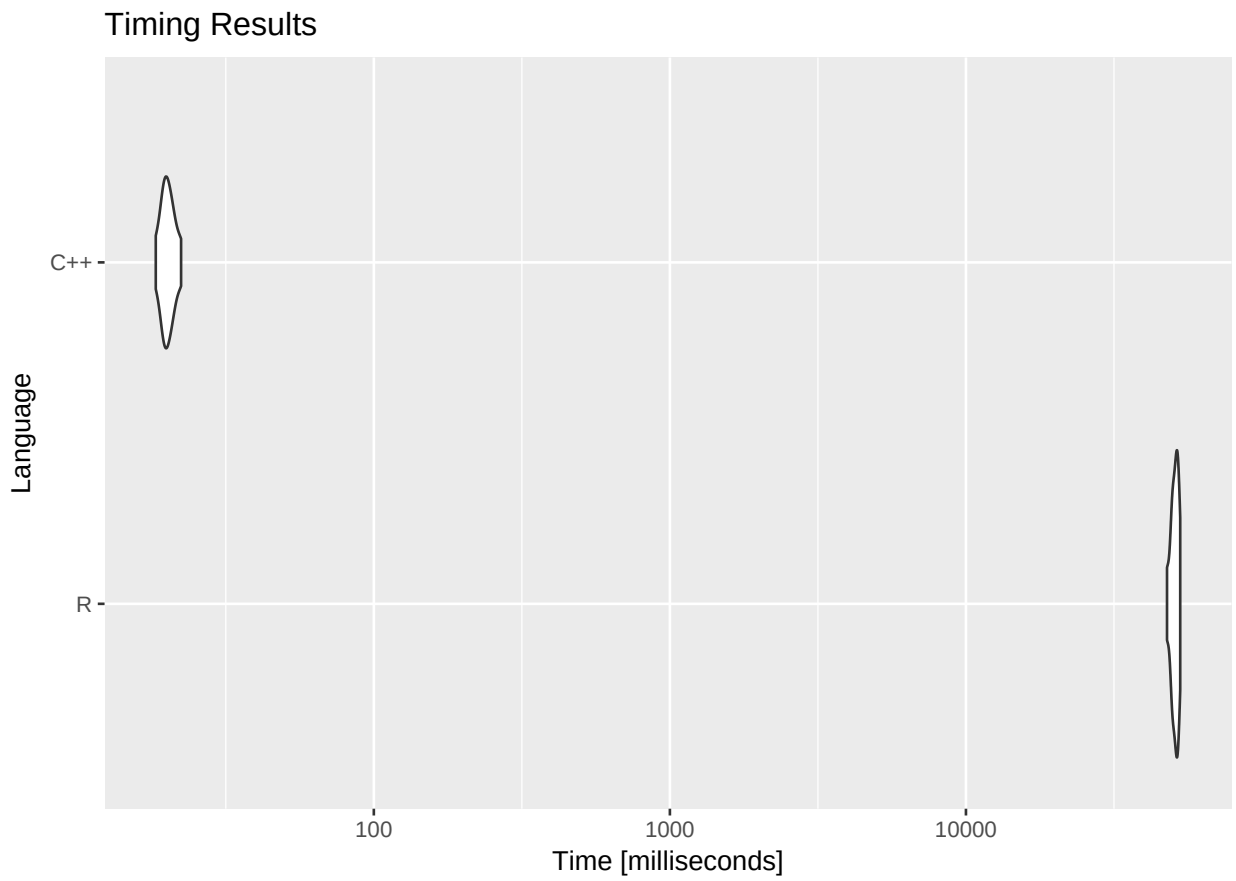
For matrices C with components:

$$C_i j = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster, and } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

These computations could become very expensive in terms of both memory space and processor time if computed literally. Instead, given our label matrices L_1 and L_2, and their corresponding C matrices, let us save time by computer our correlation as:

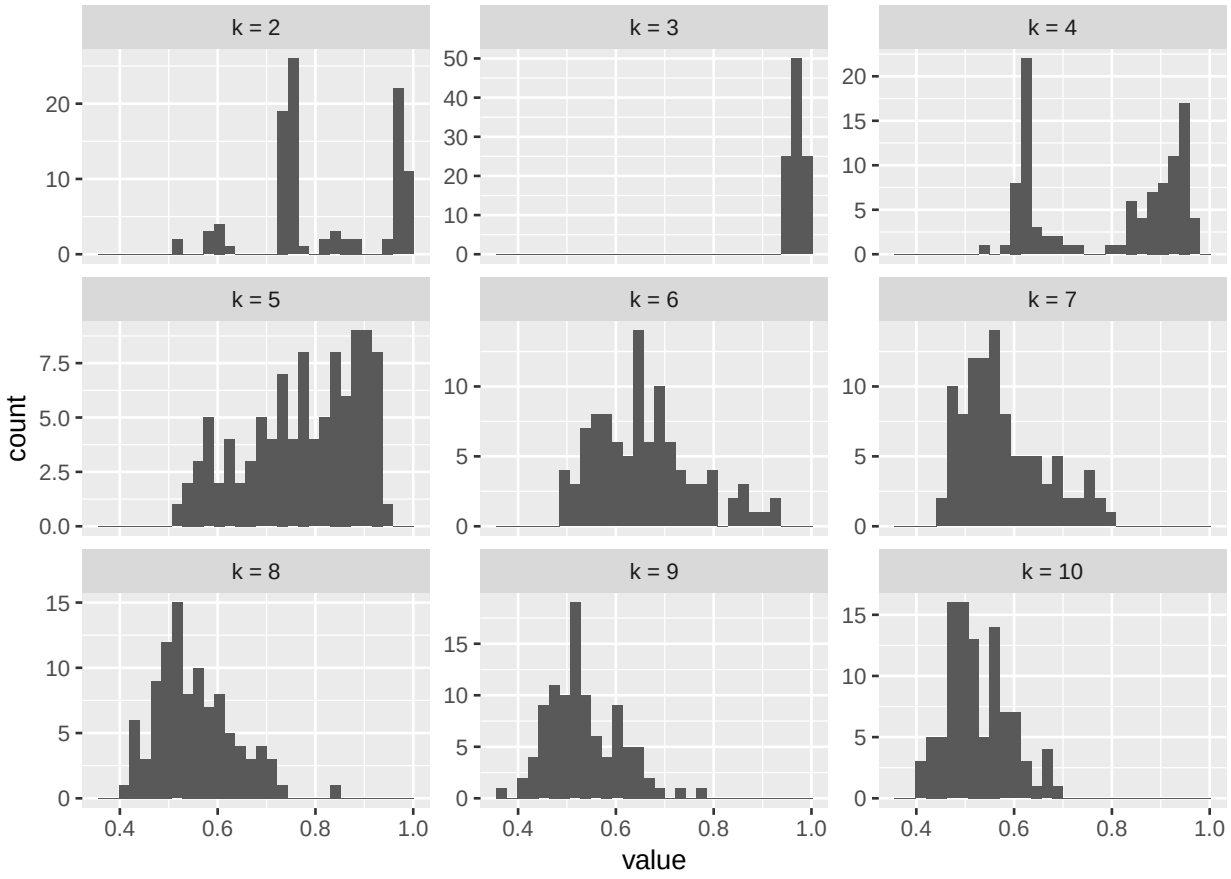
$$cor(L_1, L_2) = \frac{\sum_{i,j} C_{ij}^{(1)}, C_{ij}^{(2)}}{\sum_{i,j} C_{ij}^{(1)} C_{ij}^{(1)} \sum_{i,j} C_{ij}^{(2)} C_{ij}^{(2)}}$$

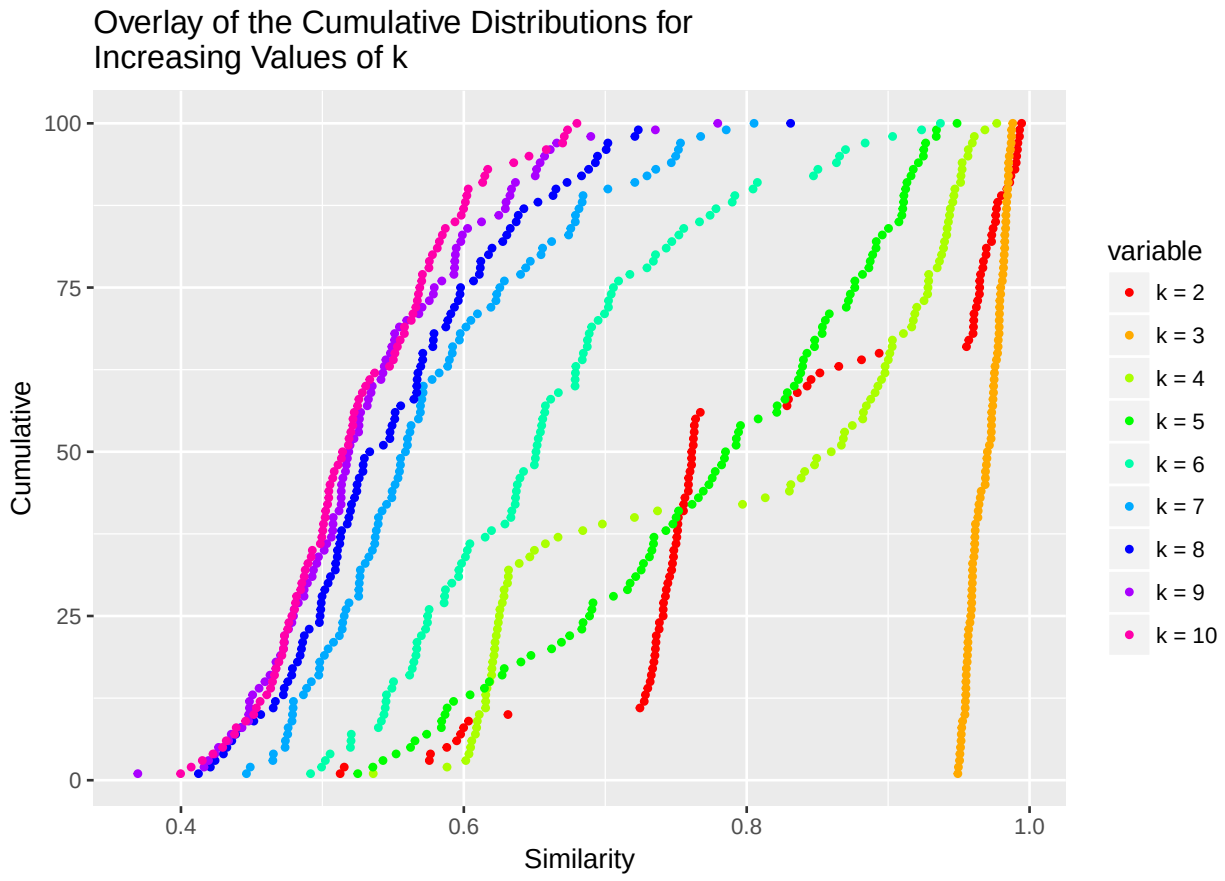
Now, not only do we prevent storing the entire $\langle L_1, L_2 \rangle$ object in memory, but it lends itself to the simple boolean/arithmetic operations for which C++ is especially well suited. But, for the same of argument, let's write this space-simplified version of the computation for both R and C++ and compare the results.



From this graph, we can see that C++ is over two orders of magnitude faster than R for almost the exact same code implementation. Given these results, it should be fairly obvious which version of the algorithm we wish to run! While many of the SCP nodes were made available for this lab, using `foreach` and my C++ function, I was able to run my analysis for $m = 0.5$, $N = 100$, and $k_{MAX} = 10$ on my 2013 MacBook Pro in under an hour. Note that this would also work perfectly well on the SCP.

2 Varied K Value Histograms and Cumulative Plot


k = 2
k = 3
k = 4
k = 5
k = 6
k = 7
k = 8
k = 9
k = 10
count
value
0.4
0.6
0.8
1.0



3 Choice of k

If we trust our cumulative plot and histograms, we can see that $k = 3$ is the optimal choice for our choice of clusters. This is the same choice as we determined in our previous lab given a less rigorous method looking at within sum of squares vs. varying values of k.

4 Discuss whether you trust the method or not

In general, I do trust this method. It might not address how appropriate k-means clustering is for a given problem, but, under the assumption that k-means is in fact the correct method to use, this is a rigorous and quantitative way to deal with the question of which value of k to choose, and how stable that choice will be in the context of the underlying distribution.

5 References

Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In Pacific symposium on biocomputing, volume 7, pages 617, 2001.

6 Acknowledgements

The author thanks Max Gardner for helpful discussions and assistance with this lab.