CPSC 320 Notes, Clustering Continued

January 26, 2015

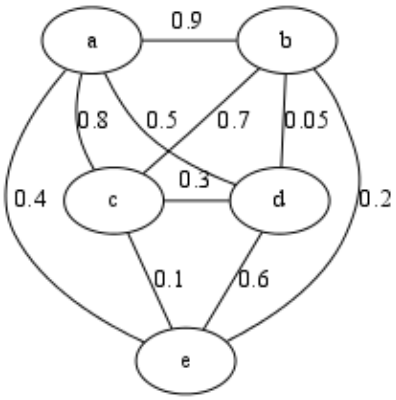
We're given a complete, weighted, undirected graph $G = (V, E)$ represented as an adjacency list, where the weights are all between 0 and 1 and represent similarities—the higher the more similar—and a desired number $1 \leq k \leq |V|$ of categories.

We define the similarity between two categories C_1 and C_2 to be the maximum similarity between any pair of nodes $p_1 \in C_1$ and $p_2 \in C_2$. We must produce the categorization—partition into k non-empty sets—that minimizes the maximum similarity between categories.

We're going to try this greedy approach as a clustering algorithm:

1. Sort a list of the edges E in decreasing order by similarity.
2. Initialize each node as its own category.
3. Initialize the category count to $|V|$.
4. While we have more than k categories:
   (a) Remove the highest similarity edge (u, v) from the list.
   (b) If u and v are not in the same category: Merge u's and v's categories, and reduce the category count by 1.

1 Practice



1. Run the greedy algorithm on this graph with $k = 2$ and write down the resulting categories.

2. What is the maximum similarity between categories in your result?

3. Imagine each similarity s replaced by $1 - s$ and then take the same steps again.

2 Analysis

1. Why is this algorithm called "greedy"?

2. Why restrict $k \geq 1$?

3. Why restrict $k \leq |V|$?

4. Prove that the algorithm terminates.

5. How long would it take to sort (by weight) the edges of an arbitrary weighted, undirected graph represented as an adjacency list using MergeSort?

6. Explain why it's not meaningful to perform an analysis in terms of both $|V|$ and $|E|$ for this problem. (Reminder: the graph is **complete**.)

7. Analyze the worst-case runtime of the algorithm in terms of $|E|$ if $k = |V| - 1$, and a (possibly loose) bound on the runtime of the second step of the loop is $O(|V|)$.

8. Analyze the worst-case runtime of the algorithm in terms of $|E|$ if $k = 1$ and the **total** cost of the second step of the loop over **all** iterations of the loop is $\Theta(\alpha(|V|))$ in the worst case. Note: $\alpha(|V|) \in \omega(1)$, $\alpha(|V|) \in o(\lg |V|)$, and $\alpha(10^{100}) < 4$.

3 Challenge Problems

Looking back at the case where $k = |V| - 1$: How could we change the algorithm to asymptotically improve this bound? Does the change have any asymptotic negative effect for any value of k? With your change, what is the worst-case runtime of the algorithm in terms of k and $|E|$?