CPSC 320 Sample Solution, Clustering Completed

February 13, 2018

AS BEFORE: 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.

Now, we'll prove this greedy approach optimal.

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 Greedy is at least as good as Anything Else

We'll start by noting that any solution to this problem partitions the edges into the "intra-category" edges (those that connect nodes within a category) and the "inter-category" edges (those that cross categories).

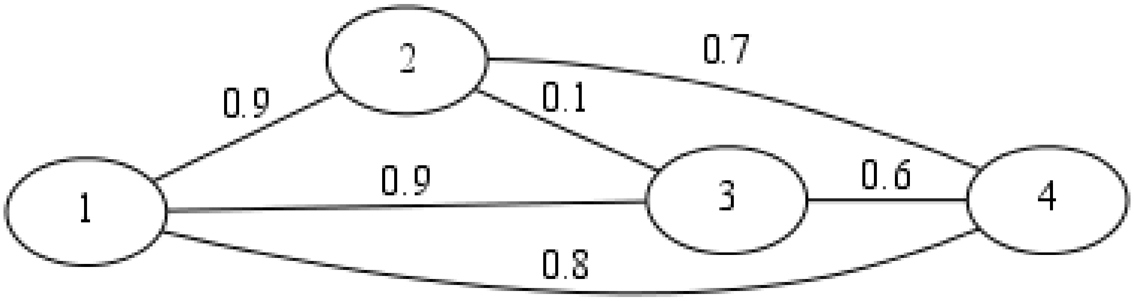
1. **Getting to know the terminology:** Imagine we're looking at a categorization produced by our algorithm in which the inter-category edge with maximum similarity is e.

   Can our greedy algorithm's solution have an intra-category edge with **lower** weight than e? Either draw an example in which this can happen, or sketch a proof that it cannot.

   SOLUTION: Can an edge between two nodes in the same category have a similarity lower than the largest-similarity edge that goes across categories?

   Why would we think this could **not** happen? Because we created the categories by merging on edges in order from highest-similarity down. However, if you've tried a few problems, you've noticed that some of the intra-category edges were never merged on. They're intra-category because a series of **other** edges leading between their endpoints all got merged.

   Let's build the smallest instance we can where there's an intra-category edge that was never merged on and then make that edge's weight low. We can get that with 2 desired categories and the graph:



$(1,3)$ and $(1,2)$ have the highest similarities and will both be merged on in 4(b). Now, we have two clusters: $\{1,2,3\}$ and $\{4\}$. Note that $(2,3)$ is intra-category, even though its weight is much lower than every inter-category edge, not just the highest-similarity one (which is $(1,4)$ at 0.8).

2. Give a bound—indicating whether it's an upper- or lower-bound—on the maximum similarity of an arbitrary solution in terms of any one of its inter-category edge weights. (That is, I tell you that the solution has an inter-category edge with weight w. How much can you tell me so far about the solution's overall "goodness"?)

   SOLUTION: The maximum similarity of an arbitrary solution is the maximum similarity of any pair of its categories, which in turn is the maximum similarity of any inter-category edge. Nothing here says that the inter-category edge we're looking at has the **maximum** similarity among all inter-category edges, however.

   So, w is not necessarily actually the maximum similarity because some other edge's weight may be larger. Even if every other inter-category edge has lower weight than w, however, the maximum similarity cannot be any **smaller** than w.

   Therefore the weight of any inter-category edge gives a **lower** bound on the maximum similarity. (I.e., max similarity $\geq w$.)

   (Neither lower- nor upper-bounds need be asymptotic bounds. For example, if you know you passed a class but you don't know the specific grade you got, you have a lower-bound on your grade of 50.)

3. Give a bound on the maximum similarity of a solution produced by the greedy algorithm in terms of the weight of any one of the edges it considered in step 4.

   SOLUTION: Since the algorithm inspects edges in order of decreasing similarity and ensures every one is intra-category (either because it's merged on in 4(b) or because it's already intra-category), no inter-category edge can have a higher weight than any edge considered in step 4.

   Thus, the weight of any edge considered in step 4 forms an **upper** bound on the maximum similarity.

4. Consider an arbitrary valid solution $\mathcal{S}$ to an instance of the problem. Prove that its intra-category edges cannot be a proper superset of the intra-category edges of the greedy solution to the same instance (i.e., cannot be the same edges plus at least one more intra-category edge).

   (**Assume** that the greedy algorithm produces valid solutions, i.e., partition V into exactly k subsets.)

   SOLUTION: Let's try a proof by contradiction. Imagine $\mathcal{S}$'s set of intra-category edges **is** a proper superset of our algorithm's. Then, everything that was intra-category before still is. (So, none of the existing categories have been "broken up" in any way.) Furthermore, because it's a *proper* superset, at least one of the existing inter-category edges must now be intra-category. That means the two otherwise intact categories on either end of (at least) one inter-category edge have now been merged into a single category. (Note that "ripping out" part of one category and moving it to the other is not an option because of our argument above that no existing categories have been "broken up".)

Thus, $\mathcal{S}$ has one fewer categories than the greedy solution. But... the greedy solution had the correct number of categories, which means $\mathcal{S}$ has the wrong number, but that's a contradiction with $\mathcal{S}$ being valid!

Therefore our assumption was wrong, and $\mathcal{S}$'s set of intra-category edges is **not** a superset of the greedy solution's.

5. Now, considering in decreasing order of weight whether each edge is inter- or intra-category, let e be the first edge where the greedy solution differs from $\mathcal{S}$. (If no such edge exists, then they have the same set of categories, are the same solution, and so our greedy algorithm is "at least as good".)

   Use the behaviour of the greedy algorithm and your work in the previous part to explain why e cannot be inter-category in the greedy solution but intra-category in $\mathcal{S}$.

   SOLUTION: Imagine this did happen. Our greedy algorithm ensures edges in this ordering are intra-category until it would run out of categories if it continued merging. Thus, this inter-category edge comes after all the edges that greedy "merged on" in step 4(b). The remaining intra-category edges in greedy are "intra" because of the ones greedy merged on in step 4(b). Since $\mathcal{S}$ agreed with greedy on the 4(b) edges, it must classify **all** of greedy's intra-category edges as intra-category itself. So, its intra-category edges are a proper superset of greedy's, but that's not possible, as we noted above. Thus, this situation cannot arise.

6. Continuing the previous part, we now know that e is intra-category in the greedy solution but inter-category in $\mathcal{S}$. Use your previous work to finish the proof that the greedy solution is "at least as good" according to our metric as the arbitrary solution $\mathcal{S}$ (and therefore the greedy solution is optimal).

   SOLUTION: Let's name the maximum similarity (the "badness") of the greedy solution $M_{\mathcal{G}}$ and of the arbitrary solution $M_{\mathcal{S}}$. We know for any solution (including $\mathcal{S}$) that any inter-category edge gives us a bound on maximum similarity: $M_{\mathcal{S}} \geq e$. As for $M_{\mathcal{G}}$, either it hasn't had an intercategory edge yet, in which case $M_{\mathcal{G}} \leq e$ and so $M_{\mathcal{G}} \leq M_{\mathcal{S}}$ **or** its highest-weight intercategory edge is above this point, in which case that is also intercategory in $\mathcal{S}$ and so $M_{\mathcal{G}} = M_{\mathcal{S}}$. Either way, $M_{\mathcal{G}} \leq M_{\mathcal{S}}$.

   Since our greedy algorithm's solution $\mathcal{G}$ is at least as good as any arbitrary solution $\mathcal{S}$, it is optimal.