# CPSC 320 Sample Solution, Clustering Completed

**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 Optimal

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).

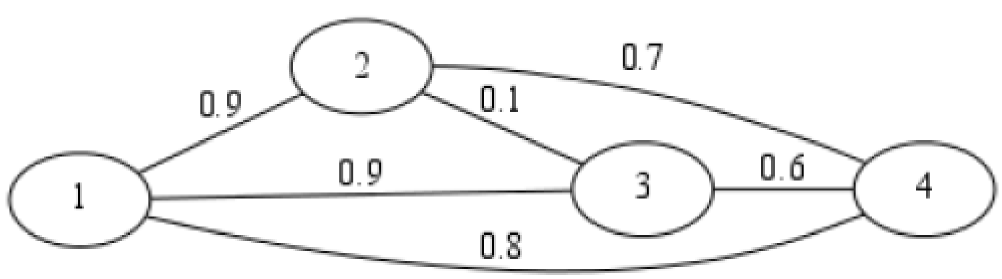
1. **Getting to know the terminology:** Imagine we're looking at a categorization produced by our algorithm in which the inter-category edge with maximum similarity is $e$.

   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 categorization $\mathcal{C}$ in terms of any one of its inter-category edge weights. That is, I tell you that $\mathcal{C}$ has an inter-category edge with weight $s$. How much can you tell me so far about $\text{Cost}(\mathcal{C})$?

   **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., $\text{Cost}(\mathcal{C}) \geq w$.)

3. Let $\mathcal{G}$ be the categorization produced by our greedy algorithm, and let $\mathcal{O}$ be an optimal categorization on that instance. Let $E'$ be the set of edges removed from the list during iterations of the While loop. With respect to the greedy solution $\mathcal{G}$, are the edges in $E'$ inter-category? Or intra-category? Or could both types of edges be in $E'$?

   **SOLUTION:** At any iteration of the While loop, if the edge $e$ removed is an inter-category edge, the categories it connects are merged and the edge becomes intra-category. So, all edges of $E'$ must be intra-category edges of $\mathcal{G}$.

4. Suppose that some edge $e = (p, p', s)$ of $E'$ is inter-category in the optimal solution $\mathcal{O}$. What can we say about $\text{Cost}(\mathcal{G})$ versus $\text{Cost}(\mathcal{O})$?

   **SOLUTION:** It must be that $\text{Cost}(\mathcal{G}) \leq \text{Cost}(\mathcal{O})$. To see why, first notice that since the algorithm considers edges in decreasing order of weight and $e$ is among the edges considered, every inter-category edge of $\mathcal{G}$ has weight at most $s$, the weight of $e$. This means that $\text{Cost}(\mathcal{G}) \leq s$. Also, since $s$ is the weight of an inter-category edge of $\mathcal{O}$, we have from part 2 that $s \leq \text{Cost}(\mathcal{O})$. Putting these two inequalities together we see that $\text{Cost}(\mathcal{G}) \leq s \leq \text{Cost}(\mathcal{O})$.

5. Suppose that all edges of $E'$ are intra-category not only in $\mathcal{G}$, but also in the optimal solution $\mathcal{O}$. Can there be any edges that are inter-category in $\mathcal{G}$ but intra-category in $\mathcal{O}$? (Hint: imagine you have a solution produced by the greedy algorithm. Can you convert any of its inter-category edges to intra-category edges without either making some edges in $E'$ inter-category or making your solution invalid?)

   **SOLUTION:** Briefly, the answer is that this **cannot** happen: the set of intra-category edges of $\mathcal{O}$ cannot contain all edges in $E'$ plus additional edges that are inter-category in $\mathcal{G}$.

   To show why, let's proceed according to the hint and try to construct a solution whose intra-category edge set includes all the edges in $E'$ plus one or more inter-category edges in $\mathcal{G}$.

   Consider an edge $(u, v)$ which is inter-category in $\mathcal{G}$, and we'll see what happens if we try to convert it into an intra-category edge. We could merge the category containing $u$ with the category containing $v$: but, this would lead to one fewer categories, which would mean our solution was invalid (because one of the requirements of a valid solution is that it must have the specified number of categories).

   Another option is to suppose that one of $u$ and $v$ – without loss of generality, let's say it's $v$ – is in a category with other nodes, and instead of merging the categories we "break" $v$ away from its category and put it into the category containing $u$. The problem with this is that, in order for $v$ to be in its current category, the greedy algorithm must, at some point, have merged on one of the edges between $v$ and one of the other nodes in its category (if this had never happened, $v$ would be in a category by itself). Therefore, **moving** $v$ into the category with $u$ means we will "lose" at least one of the edges in $E'$ (i.e., it will become inter-category in the new solution).

Therefore, if $\mathcal{O}$ is a valid solution (which it must be, or else it wouldn't be optimal) in which all edges in $E'$ are intra-category, it cannot be the case that any of its intra-category edges are inter-category in $\mathcal{G}$.

6. Apply the progress made in parts 3 to 5 to conclude that $\mathcal{G}$ must be an optimal solution.

   **SOLUTION:** Based on questions 4 and 5, we consider the proof as two separate cases: all the edges of $E'$ are intra-category in $\mathcal{O}$, or **not** all the edges of $E'$ are intra-category in $\mathcal{O}$.

   In the first case (all edges in $E'$ are intra-category in $\mathcal{O}$), we have by part 5 that none of the inter-category edges in $\mathcal{G}$ are intra-category in $\mathcal{O}$. Therefore, the inter-category edges in $\mathcal{O}$ are a superset of the inter-category edges in $\mathcal{G}$, so $\text{Cost}(\mathcal{G}) \leq \text{Cost}(\mathcal{O})$. (Technically, in this case we actually have that $\mathcal{O}$ is the **same solution** as $\mathcal{G}$ and therefore the costs are the same; this is not too difficult to show, but it isn't actually necessary for the proof.)

   In the second case (not all edges in $E'$ are intra-category in $\mathcal{O}$): by part 4, we know that in this case $\text{Cost}(\mathcal{G}) \leq \text{Cost}(\mathcal{O})$.

   Therefore, in either case, $\text{Cost}(\mathcal{G}) \leq \text{Cost}(\mathcal{O})$, which completes the proof that $\mathcal{G}$ is optimal.