CPSC 320: Clustering Solutions (part 2) *

Step 5 Continued: Correctness of Greedy-Clustering

Our goal is to show that the greedy algorithm we developed in the previous worksheet for our photo clustering problem (reproduced below) produces a categorization that minimizes $\text{Cost}(\mathcal{C})$. Recall that an instance of the problem is

- n, the number of photos (numbered from 1 to n);
- E, a set of weighted edges, one for each pair of photos, where the weight is a similarity in the range between 0 and 1 (the higher the weight, the more similar the photos); and
- c the desired number of categories, where $1 \leq c \leq n$.

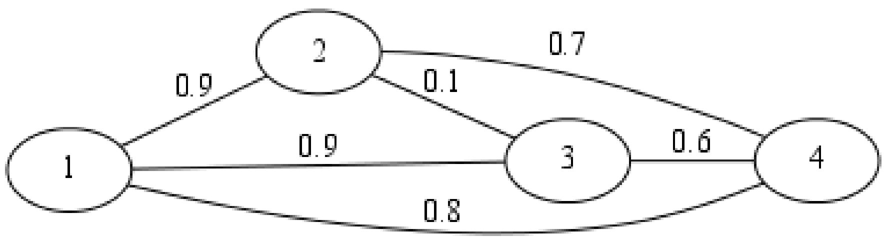
A *categorization* $\mathcal{C}$ is a partition of the photos into c (non-empty) sets, or categories. If $\mathcal{C}$ has more than one category, the *inter-category* similarity between two of its categories C_1 and C_2 is the maximum similarity between any pair of photos $p_1 \in C_1$ and $p_2 \in C_2$. Edges between photos in the same category are called *intra-category* edges. The *cost* of the categorization is the maximum inter-category similarity, taken over all pairs of categories. We'll denote the cost by $\text{Cost}(\mathcal{C})$. The lower the cost, the better the categorization, so we are trying to find the categorization with minimum cost.

```
function CLUSTERING-GREEDY(n, E, c)
▷ n ≥ 1 is the number of photos
▷ E is a set of edges of the form (p, p′, s), where s is the similarity of p and p′
▷ c is the number of categories, 1 ≤ c ≤ n
    create a list of the edges of E, in decreasing order by similarity
    let C be the categorization with each photo in its own category
    Num-C ← n              ▷ Initial number of categories
    while Num-C > c do
        remove the highest-similarity edge (p, p′, s) from the list
        if p and p′ are in different categories of C then
            "merge" the categories containing p and p′
            Num-C ← Num-C − 1
    return C
```

1. We'll start by getting to know the terminology. Imagine that we're looking at a categorization produced by our algorithm in which e is the inter-category edge with highest similarity. 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: Why might we think that there is no such intra-category edge? Because we created the categories by merging on edges in order from highest-similarity down. However, if you've tried a few instances, you may have noticed that some of the intra-category edges were never merged on. They're intra-category because a series of **other** edges connecting them all got merged.

   Let's find a counterexample, by building 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 two desired categories (i.e., $c = 2$) and the graph:

   In this graph, $(1, 3)$ and $(1, 2)$ have the highest similarities and so after the first two steps, photos 1, 2, and 3 will be in the same category. 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.

2. Suppose that I tell you that $\mathcal{C}$ has an inter-category edge e with weight s. Can you find an upper bound or lower bound on $\text{Cost}(\mathcal{C})$ in terms of s?

   SOLUTION: The maximum similarity of $\mathcal{C}$ is the maximum similarity of any inter-category edge. Nothing here says that e has the highest similarity among all inter-category edges, however.

   So, s 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 s, however, the maximum similarity cannot be any smaller than s.

   Therefore the weight s of any inter-category edge gives a *lower bound* on the maximum similarity of $\mathcal{C}$. That is, $\text{Cost}(\mathcal{C}) \geq s$.

3. On to proof of correctness of our greedy algorithm. Fix an instance of the problem. In what follows, let $\mathcal{C}^G$ be the categorization produced by our greedy algorithm, and let $\mathcal{C}^*$ 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 $\mathcal{C}^G$, 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 $\mathcal{C}^G$.

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

   SOLUTION: It must be that $\mathrm{Cost}(\mathcal{C}^G) \leq \mathrm{Cost}(\mathcal{C}^*)$. 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 $\mathcal{C}^G$ has weight at most s, the weight of e. This means that $\mathrm{Cost}(\mathcal{C}^G) \leq s$. Also, since s is the weight of an inter-category edge of $\mathcal{C}^*$, we have from part 2 that $s \leq \mathrm{Cost}(\mathcal{C}^*)$. Putting these two inequalities together we see that $\mathrm{Cost}(\mathcal{C}^G) \leq s \leq \mathrm{Cost}(\mathcal{C}^*)$.

   (Intuitively, by ensuring that high-weight edges are intra-category edges, the greedy algorithm "stays ahead" of the optimal solution $\mathcal{C}^*$.)

5. Suppose that all edges of E' are intra-category not only in $\mathcal{C}^G$, but also in the optimal solution $\mathcal{C}^*$. Can $\mathcal{C}^G$ and $\mathcal{C}^*$ be different?

   SOLUTION: No: $\mathcal{C}^G$ must be equal to $\mathcal{C}^*$. This is because every category of $\mathcal{C}^G$ must be a subset of a category of $\mathcal{C}^*$. Intuitively, this is because every decision made by the Greedy algorithm to merge categories is consistent with $\mathcal{C}^*$, since all edges in E' examined by the algorithm are intra-category in $\mathcal{C}^*$. But then, since $\mathcal{C}^G$ and $\mathcal{C}^*$ have the same number of categories, two different categories of $\mathcal{C}^G$ can't be subsets of the same category of $\mathcal{C}^*$. Rather, $\mathcal{C}^G$ and $\mathcal{C}^*$ must have identical categories.

   [Note: We could argue more formally, using induction, to show that every category of $\mathcal{C}^G$ must be a subset of a category of $\mathcal{C}^*$. The induction argument is on iterations of the While loop of the Greedy algorithm. Suppose that $\mathcal{C}_i$ is the categorization after i iterations of the While loop. The base case is when $i = 0$: In $\mathcal{C}_0$ every photo is in its own category, and so is trivially a subset of a category of $\mathcal{C}^*$. Let $i \geq 1$ and suppose (induction hypothesis) that every category in $\mathcal{C}_{i-1}$ is a subset of some category in $\mathcal{C}^*$. For the induction step there are two cases. The first case, that $\mathcal{C}_i = \mathcal{C}_{i-1}$, is trivial. The other case is that two categories of $\mathcal{C}_{i-1}$, say C and C' containing photos p and p' respectively, are merged to form a single category $C \cup C'$ of $\mathcal{C}_i$, making edge $e = (p, p', s)$ intra-category. Then, since edge e is in E', p and p' must be in the same category of C^*. So, C and C', and thus $C \cup C'$ must be subsets of the *same* category of $\mathcal{C}^*$.]

6. Apply the progress made in parts 4 to 5 to conclude that $\mathcal{C}^G$ must be an optimal solution.

   SOLUTION: Let $\mathcal{C}^*$ be an optimal solution. There are two cases: either the set E' of edges considered by the greedy algorithm are all intra-category in $\mathcal{C}^*$, or some edge of E' is inter-category in $\mathcal{C}^*$. In the former case, by part 5, $\mathcal{C}^G = \mathcal{C}^*$ and so $\mathcal{C}^G$ is optimal. In the latter case, by part 4, $\mathrm{Cost}(\mathcal{C}^G) \leq \mathrm{Cost}(\mathcal{C}^*)$, and since the goal is to minimize cost, again we can conclude that $\mathcal{C}^G$ is optimal.