[Return to visualization]

Author: Maks Cegielski-Johnson

Project can be found at *https://makscj.github.io/adv-vis/proj2/*

- ## Purpose

  The purpose of this project is as follows:

  Develop an algorithm (or use existing algorithms) to identify the set of n Utah Census Tracts in k contiguous clusters (having a minimum cluster unemployment rate of 6.5 percent) that maximizes the total population across all n Census Tracts.

  The federal government has special funds to help local areas with substantial unemployment (ASUs).

  An ASU is defined as a cluster of Census Tracts with a cluster unemployment rate of at least 6.5 percent.

  Federal funding for programs to help these areas is distributed to the states based on the total population of the ASUs in the state

  The state of Utah would like to identify the set of ASUs with maximum total population in order to maximize federal funding.

- ## Visualization

  The map provides a visualization of all of the tracts in Utah. Clicking on any tract will add it to the consideration for clustering in an ASU. ASUs are automatically created through interaction with the map, that is, any adjacent tracts that are added to consideration by the user are added to the same ASU, and each unique ASU is automatically assigned some categorical color for easier identifaction.

  All of the ASUs are listed to the side of the map, along with the corresponding color and the current unemployment rate of each ASU. Any ASU that has an unemployment rate below the 6.5 percent threshold are marked with red font. **Clicking on a color icon** will zoom the map into the clicked ASU to allow further inspection. The user can also freely zoom in and out of the map as well as pan around.

  Hovering over a tract which is not considered in any ASU will display it's population and unemployment rate. Hovering over a tract in an ASU will display the total population of that ASU aswell as the unemployment rate of that ASU.

  The **Initial Seeds** button will identify all tracts that have an unemployment over 6.5 percent and automatically cluster them into ASUs.

- ## Solution

  My solution is currently just identical to what the state is doing.

  The state identifies which tracts, on their own, meet the unemployment greater than 0.065 criteria to be an ASU.

  All of these tracts will be included in a maximal solution.

  They might be what you would consider your *seed* tracts.

Once those are mapped, we can expand out a contiguous cluster (using queen contiguity) around each seed tract.

The unemployment rate and total population must be re-calculated with each addition to make sure the cluster does not drop below 6.5 percent.

If it does we remove that tract (or a different one if it results in higher total population).

Explanation

I believe this problem is at least NP Complete, if not NP Hard, therefore I think that this solution is reasonable. Because of the nonlinear update step required for each unemployment rate, I couldn't find a method to construct an ILP. I also could not come up with a heuristic that would allow me to use A Star, because simply adding the unemployment rate of a single tract to the rate of an ASU is not admissable.

Furthermore, because the unemployment rate update is not simple, there is no way to know if the the current solution is optimal without trying every combination of ASUs, and is therefore at least NP Complete. Consider the following theoretical tracts

	A	**B**	**C**
u	128	1	67
u+e	1000	1000	1000
ur	0.128	0.0	0.067

Where A is a neighbor with B, and B is a neighbor with C, but A is not a neighbor of C.

A and B will not be joined because their combined rate is 0.0645 which is below the threshold. Similarly, B and C will not be joined because their rate is 0.034.

However, consider the combined rate of {A, B, C}. Their rate is 0.065333, which is above the threshold, and therefore is a legal ASU. Also, the maximal solution would be all three tracts, but the algorithm would only consider the solution to be {A} and {C}

Hence, I believe this example could be extended to a proof which would state that each combination of tracts must be every sigle combination of tracts, and this search space is exponential in Big Oh, therefore there does not exist an exact algorithm that we can efficiently run. So we must use an approximate algorithm for the solution, and this one seems to perform reasonably, especially since our search space is rather small (only 588 tracts).

Given more time, an extension of the algorithm I might consider is to allow some exploration to add non optimal tracts to an ASU to allow exploration of other candidate tracts. However, I can imagine that this extension would by-itself increase the computation time.

- ## Implementation

This assignment had a lot of unforseen problems that increased the difficulty. I spent a day or two trying to find an algorithm that would given me an adjaceny matrix of all of the neighbor tracts, however there seems to be a problem with the Shapefile that causes any methods I could find to break. In the end, I was able to find a csv file online which listed all of the tracts in the US, which I filtered down to only Utah neighbors.

After this, I got stuck by misinterpreting the problem, thinking that the goal was to find one population maximizing ASU, and not all of the clusters.