Monte Carlo Simulation Experiment

Scientific / Statistical Question

I am exploring whether or not it is possible to create meaningful groupings of coffee consumers through online survey data. I am looking to explore the limitations of the Proximus clustering algorithm through a controlled Monte Carlo simulation experiment. Proximus takes logical matrices as input, and produces clusters assignments for each individual observation, with the quality of the data greatly affecting the output of the algorithm.

In real-world settings, two majors issues with business data is sample size and the quality of the data. Following a factorial experimental design, I will explore how sample size and logical matrix sparsity affect the output of the Proximus clustering algorithm.

Data

- I will be generating new data based on the historic distribution of each column, with a normalized error, for each instance of a factor-level simulation. I will end up with 90 randomly generated, similar data to the current survey results, each with different amount of observations or sparistiy depending on the factor-level specification.

Estimates

- Jaccard similarity of the total approximation is provided for each proximus output, and I will be comparing the mean and standard deviation of the Jaccard similarity for each factor-level.
- The Jaccard similarity of the total approximation is defined as the sum of a procedure done to each column pair in a matrix. The procedure is as follows: The size of the intersection of the columns divided by the size of the union of the columns, ranging from 0 (no similarity) to 1 (identical columns).

Methods

I will be evaluating the performance of the Proximus algorithm from the Clustering for Business Analytics R package by Christian Buchta. I will be exploring how this algorithm handles data matrices with a differing number of observations, and how logical matrix sparsity effects the cluster output.

Performance Criteria

- The algorithm will produce the Jaccard similarity of the total approximation for each of the 90 instances of a simulation. A value of 1 indicates that the 0's and 1's of the original logical matrix were perfectly preserved (i.e. unchanged from the original). When parameters that affect algorithmic performance are changed, I am expecting to see shifts in this parameter as the algorithm begins to perform worse.

Simulation Plan

Differences from Project III:

- Project III Utilized Grower/PAM clustering to attempt to group consumers based on their reported survey responses, however we were unable to uncover well defined clusters, which made it difficult to group individuals based on their responses. For Project IV, I will be using the Proximus clustering algorithm, and have converted survey responses into a logical matrix.

Simulation:

- Our two factors are the number of observations, and the logical matrix sparsity. Each simulation will use a different combination of factor-levels, leading to 9 factor-level simulations total.
- Each factor-level simulation will have 10 simulations with randomly generated data following the factor-level parameters (low obs-low sparsity, low obs-medium sparsity, etc). This will lead to 90 simulations total.
- The Factors will have the following levels:
  - Number of Observations: low (500), medium (1500), high (2790)
  - Logical Matrix Sparsity: low (1 columns with 90% sparsity), medium (3 columns with 90% sparsity), high (6 columns with 90% sparsity).

- The Proximus model output will be created and stored for each factor level, for each of the simulations. As a result we will have 90 Proximus algorithm outputs, with 10 per factor-level combination.

Anticipated Challenges or Limitations

In the event that individual observations are too closely related due to observed similarities in the logical matrix, Proximus will likely perform poorly in all 9 factor-level simulations. This is because there would be an extreme lack of differences in the "distance" between each observation, preventing unique clusters to be formed. In this case, Proximus would not be the best fit for our data, and we would likely see little to no difference between each output.