MGGG- Colorado

Precinct Boundaries & Geocoding

The ongoing efforts to implement a gerrymandering assessment tool for Colorado additionally face the challenge of having to validate the produced results, particularly the assignation of votes to the right precinct.

An initial attempt of bulk geocode the information from the addresses obtained from the SoS was made with an in-house bulk-geocoding code utilizing an ArcGIS API for Python (<https://developers.arcgis.com/python/>). The API is limited in volume (1,000,000 free queries per month <https://developers.arcgis.com/pricing/>) and in rate (about one query per second). However, experimentally the rate observed was substantially better at around 1.7 queries per second, to even 60 per second with larger files (with timeout=None). Files of up to 40,000 addresses were successfully geocoded, with a very low 0.33% failure rate (returning no points) though with accuracy that is hard to assess.

A preliminary assessment method for the quality of the geocoding was performed. The basic approach was to reverse geocode the coordinates generated, and later compare the address obtained from that process to the original input. The comparison method was a simple string compare. The accuracy assessment method is limited as it was shown that ill-scored results were geocoded correctly (think of corner houses). Additionally, the reverse geocoding method has the large issue of not being able to “guess” apartment and unit numbers. The string method sorts the data in terms of “quality” though accuracy is harder to assess. A last concern is the accuracy of the reverse geocoding itself, which is assumed to be at least as good as the geocoding. Since both are from the same API it seems like a reasonable guess, though perhaps a third-party validation (ie different API) would be a more sensible approach.

The selection the API is justified by its large permitted volume, cost and ease of use. However, forums express that this API is not the most accurate one, with GoogleV3 claiming that title. I have not assessed the accuracy of GoogleV3 but even if it was substantially more accurate, it still doesn’t make up to the fact that it is limited to 2,500 queries per day, and 0.5$/1000 thereafter. The task at hand might present an issue of balancing volume and accuracy. An idea for this would be to perform the bulk geocoding with the ArcGIS API and then “sieve” out the poorer results (by some quality assessment) and re-perform geocoding on those with the GoogleV3 API.

A free, publicly-available web application from the US Census Bureau with batch capabilities (<https://geocoding.geo.census.gov/>) was also assessed yielding the following results:

Out of 10,000 attempted (maximum batch allowed, no limit on number of batches):

5,445 successful addresses (as reported by the API)

2669 non-exact addresses (as reported by the API)

1886 empty returns.

Success rate: 81% (max), 54% (min)

Time elapsed: 34 min (294 address/minute).

Note that the non-exact results are largely cases like apartments and corner cases, making it not a bad option. It however, is limited by its rigidity in its desired input format, in which additional attributes (like precinct ID) are not accepted. Given than the input requires indexing this problem can be navigated by the later addition of the desired attributes, for which the custom code df\_handling has been produced.

The recommendation after this initial attempt is to use the Census API, sieve out all the empty results and use the ArcGIS API which has a much larger success rate.