ReadMe for gc_cover_git_v4 iPython notebook

Jeff Nivitanont, University of Oklahoma

April 24, 2018

1 Introduction

Packages:

- netCDF4 - used to read in netCDF4 formatted files
- NumPy - part of the SciPy suite, provides data structures such as n-d arrays and series
- pandas - Python Data Analysis library provides easy to read and manipulate DataFrame
- shapely - Python wrapper for C-GEOS (Geometry Engine - Open Source), which is a port of the JTS (Java Topology Suite)
- GeoPandas - Geospatial Data Analysis library built from pandas
- matplotlib - Primary plotting library for python
- CartoPy - add-ons for plotting geospatial information in matplotlib
- descartes - add-ons for plotting polygons in matplotlib

We are given the problem of optimizing the scan pattern for the GeoCARB satellite instrument set to launch in 2022. GeoCARB is a geostationary satellite that will be positioned at $0° - 85°W$ and make daily observations of atmospheric CO2, CO, NH4, and SIF over the Americas. The underlying mathematical problem related to optimizing the scanning pattern is called the Geometric Set Cover problem.

> *Given a set of points in 2-D space and a set of covers, what is the optimal covering set?*

This problem is known to be NP-Hard and there are no efficient methods for straight-forward computation. However, there are heuristic methods we can utilize for finding a set cover, though they do not necessarily yield optimal covers. We explore this problem with the Greedy Heuristic Algorithm in the iPython environment.

2 Cover Set

We use the shapely package to create the scan block Polygons from the coordinates of the corners from the scan blocks indicated in the netCDF4 file. We plot them on our map, shown in Figure 1 using the matplotlib and CartoPy libraries.

3 Area of Interest

Through a mathematical lens, we can consider the area of Earth lying within the square $[50°N, 50°S] \times [-130°W, -30°W]$ as our Universe Set, $\mathbb{U}$. The land within our Universe Set, excluding the Caribbean Islands, is our area of interest which we will call $\mathbb{E}$. We will consider the ocean and the Caribbean as $\mathbb{E}^c$ so that $\mathbb{E} \cup \mathbb{E}^c = \mathbb{U}$.

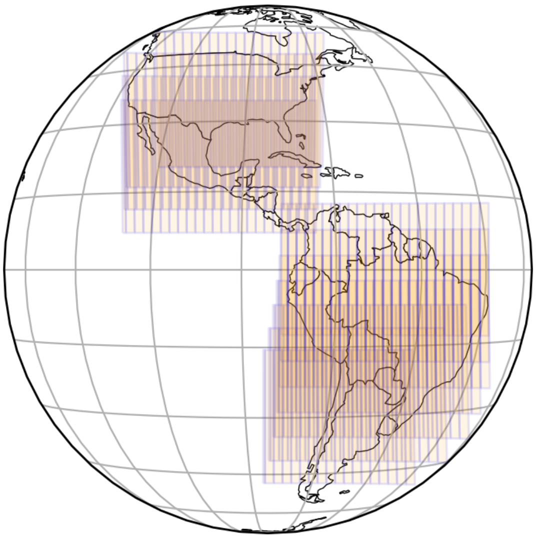

Figure 1: The set of covers.

3.1 Implementation

Using a built-in GeoPandas dataset we select North America and South America geometries for our set to be covered. The Data is 1:110m from NaturalEarth and includes cultural and physical information (i.e. countries, pop, geometries, etc.). NaturalEarth data is a free for use and supported by NACIS (North American Cartographic Information Society). After obtaining our geometries, we truncate our set to only include land between 50N and 50S and exclude eastern Nova Scotia, which lies outside our scanning blocks, shown in blue in figure 2.

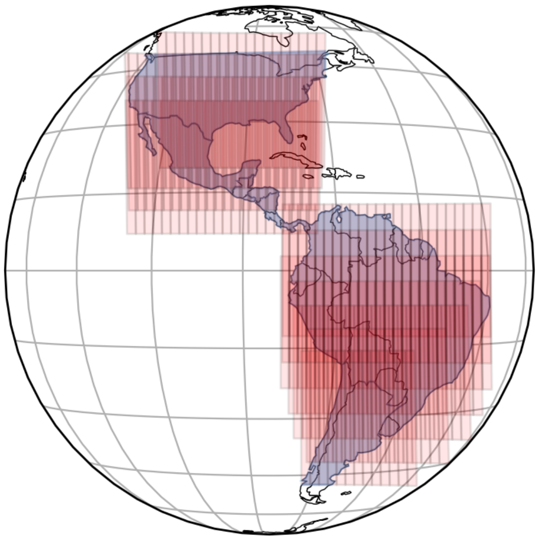

Figure 2: Area of Interest in blue.

4 Selection

4.1 Airmass Factor

To optimize our scan pattern, we look to a metric that we call the Airmass Factor, abbreviated AF. AF is deterministic and used directly in the calculation of Signal-Noise Ratio in the retrievals, so looking at AF can give us a sense of retrieval quality without the other random factors i.e.

aerosols, albedo, etc. The formula to calculate Airmass Factor is,

$$AF(lat, lon, time) = \frac{1}{SolarZenithAngle} + \frac{1}{SatelliteZenithAngle}$$

We consider Airmass Factors greater than 5 to be a poor quality retrieval, so we omit land masses whose daily minimum exceeds this value from our Universe Set.

To calculate the Airmass Factor of a scan block, we overlaid a grid of lat/lon points in $\mathbb{U}$ at half-degree intervals. We calculate the AF at each grid point that intersects a block and use the average of those points as the AF for a scan block.

4.2 Cost Function

Selection within the our Greedy Algorithm is dictated by the cost of each scan block given the time. Our current cost function is as follows:

$\mathbb{E}$ = Land Mass
$\mathbb{E}^c$ = Ocean
$\mathbb{I}$ = Selected Scan Blocks
δ = Distance from last selected scan block.
$AF(s,t)$ = Average Airmass Factor of a scan block with respect to time.

$$c(s,t) = \exp(AF(s,t))(1 + \frac{(s \cap \mathbb{I}) + \delta^2 + (s \cap \mathbb{E}^c)}{s \cap \mathbb{E}})$$

4.3 Greedy Algorithm with Cost Function

With the established area of interest, set of subcovers (scan blocks), and cost function, we are ready to implement the Greedy Algorithm for finding covering sets.

$\mathbb{E}$ = Land Mass
$\mathbb{E}^c$ = Ocean
$\mathbb{S}$ = Set of Candidate Scan Blocks
$\mathbb{I}$ = Selected Scan Blocks
$\mathbb{C}$ = Cost Function
begin
  $\mathbb{I} = \emptyset$
  while($\mathbb{E}$ not empty**):**
    $\mathbb{L} = s_i \in \mathbb{S}$, such that $\mathbb{C}(s_i)$ is the minimum of $\mathbb{C}(\mathbb{S})$
    $\mathbb{I} = \mathbb{I} \cup \mathbb{L}$
    $\mathbb{E} = \mathbb{E} \setminus \mathbb{L}$
  return $\mathbb{I}$

5 Results

After running the algorithm, we obtain a covering set from the algorithm stored in a GeoPandas.GeoDataFrame plotted in red in Figure 4.

5.1 Error

To gain a sense of "success" for a covering set, we look towards the distribution of uncertainty, or error, for each point on the Airmass grid see in Figure 3.

6 Details

More detailed descriptions of the code can be found in the iPython notebook.

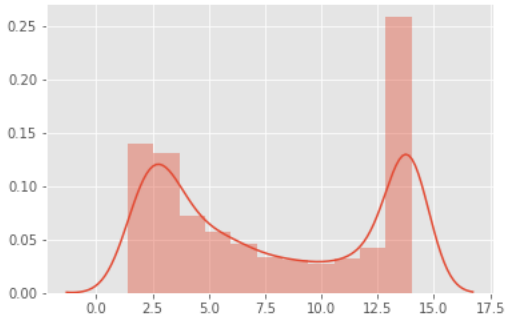

Figure 3: Distribution of Measurement Uncertainty for a 2007-03-20 Covering Set.

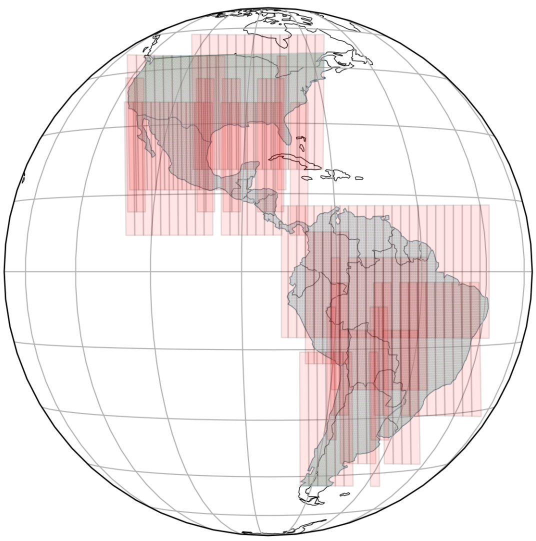

Figure 4: Scan blocks selected by algorithm.