Aftershocks

Eduard Campillo-Funollet

Earthquakes are caused by the sudden release of energy initiated at a rupture below the surface. After an initial earthquake, the *mainshock*, the region surrounding the initial rupture might be unstable, causing secondary earthquakes, the *aftershocks*. We will study a dataset of earthquakes, and model the probability of aftershocks based on quantities such as the distance to the mainshock rupture.

Question 1

We have several tables with information about earthquakes. `all_events.csv` contains the `date`, location (latitude `lat` and longitude `lon`), identifier `id`, intensity `mw` and seismic moment `moment` of many earthquakes. The tables in the folder `aftershocks/` contain the mechanical stresses `s1,...,s6` at different locations surrounding a mainshock, and a column indicating if an aftershock was identified at that location (0 if aftershock was not recorded, 1 otherwise). The table `selectedEvents.csv` contains a list of identifiers `id` and a list of the files with the corresponding aftershock tables.

(a) Create a new dataframe with four columns: `date`, `file`, `lat`, `lon`, `mw`, `aftershocks` with a row for each of the selected events, containing the date (from `all_events.csv`), the file containing the aftershock information (from `selectedEvents.csv`), the location of the mainshock, the intensity and the total number of aftershocks. Make sure the new dataframe is sorted by date, and display the first few rows using `head`.

(b) Implement a function `process_stress(fi, fu)` that receives the name of an aftershock file `fi` and a function `fu`. `fu` receives six arguments (the stress components `s1,...,s6`), and returns a single value. `process_stress` returns a data frame with columns `x, y, fu` and `aftershock`, with values from the corresponding aftershock file, and the outputs of the function `fu` for each row. Apply it to the event `2001BHUJIN01YAGI` with $f(s_1, \ldots, s_6)$ = _i |s_i|$, and display the first few rows of the result with `head`.

(c) Create new dataframe with four columns, `file` (from `selectedEvents.csv`), `lat`, `lon`, and `moment` (from `all_events.csv`). Sort it by the column `file` and display the first few rows with `head`.

Question 2

Note: if you are not familiar with any of the *geoms* required for this question, check the documentation of `ggplot` or `plotnine`, either with the RStudio help or searching the online documentation.

(a) Use `geom_map` (Python) or `geom_sf` (R) and the file `worldMap.shp` to plot a map of all the events in `all_events.csv`, a point for each event. Note: in R, you will need to read `worldMap.shp` first using the function `st_read` from the library `sf`; in Python, read `worldMap.shp` using `geopandas.read_file`.

(b) Use `geom_map` (Python) or `geom_sf` (R) with `worldMap.shp` to plot a map with a point for each event in `selectedEvents.csv`. Use colour to represent the intensity, and size to represent the number of associated aftershocks. Note: in R, you will need to read `worldMap.shp` first using the function `st_read` from the library `sf`; in Python, read `worldMap.shp` using `geopandas.read_file`.

(c) Plot the Euclidean norm of the stresses for `2001BHUJIN01YAGI` at the (x, y) coordinates in the corresponding file, using colour for the value of the norm, and include black points at the location of the aftershocks.

Question 3

We are going to model the probability of an aftershock with

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}, \tag{1}$$

where x will be a variable that we use to make the prediction. We are going to find the *best* parameter values $\beta_0, beta_1$ to model the data of a given main event, by finding the values of β_0, β_1 that minimise

$$f(\beta_0, \beta_1) = \sum_k -y_k \log(p(x_k; \beta_0, \beta_1)) - (1 - y_k) \log(1 - p(x_k, \beta_0, \beta_1)). \tag{2}$$

This expresion corresponds to the negative log-likelihood of a model. Here $y_k \in \{0, 1\}$ is the observed outcome (no aftershock or aftershock present), and x_k is our *predictor* variable, that we will define based on information about the earthquake.

(a) Implement a function `fit(X,Y,gamma)` that receives the vectors with values x_k and y_k, and a step `gamma` for the gradient descent method, and returns β_0, β_1 obtained the gradient descent method with starting point $(0, 0)$. Test it by computing the values for `TODO` using the Euclidean norm of the stresses as X and the value of the column `aftershock` as Y.

(b) Implement a function `fit_file(fi,fu,gamma)` that finds the optimal values of β_0, β_1 using gradient descent as before, using the data in the aftershock file `fi`, and the function `fu` on the stresses (defined as in Question 1b). Test it by computing the values for `TODO` using the Euclidean norm of the stresses as X and the value of the column `aftershock` as Y.

(c) Implement a function factory `fit_file_factory(fu,gamma)` to fix the values of `fu` and `gamma` in `fit_file`. Compute the values of β_0, β_1 for all events in *selectedEvents*, using $f(s_1, \ldots, s_6) = \log(\sum_i |s_i|)$ and $gamma = 10^{-3}$. Plot the results with β_0 in the x-axis and β_1 in the y-axis, one point for each event.

Question 4

The logistic regression model from Question 3 can be extended to more variables, by defining the probability

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}. \tag{3}$$

(a) Write a function `moment_distance(fi)` that receives the name of an aftershock file, and returns a dataframe with three columns: the mainshock seismic log-moment (log of `moment` in `all_events.csv`), the distance between the mainshock and the possible aftershock location computed (assume that the mainshock is at the centre of the grid of points in the aftershock file), and column with the presence/abscence of an aftershock. Use the column names `moment`, `distance`, `aftershock`, and note that the `moment` is the same for all the rows, since we are looking only at one mainshock event. Display the first few rows of the dataframe obtained by applying this function to `TODO`.

(b) Implement a function `fit2(X1,X2,Y)` that minimises the negative log-likelihood function f in Question 3 and returns the values of $\beta_0, \beta_1, \beta_2$. Use `optim` (in R) or `scipy.optimize.minimize` in Python, and **do not** use the derivative of f. Obtain the values of $\beta_0, \beta_1, \beta_2$ for `2001BHUJIN01YAGI` using `moment` for x_1, `distance` for x_2 and `aftershock` for y.

(c) Implement a function `fit2_file(fi)` that returns the values of $\beta_0, \beta_1, \beta_2$ for the aftershock file `fi` using `moment` for x_1, `distance` for x_2 and `aftershock` for y. Plot the values of β_1 vs β_0 and β_2 vs β_0 in two separate plots, one point for each event in `selectedEvents.csv`.