Weather Generator

August 7, 2020

Weather Generator is learning project consisting of a very simple stochastic algorithm for predicting weather from datasets, using probability distributions and single exponential smoothing.

Algorithm used is quite simple, use simple mathematicals concepts and take into consideration only a few parameters, so the efficiency is relative and only allows a globally consistent prediction but without extreme scenarios.

Contents

Part I
Algorithm

1 Predicting the day's temperature of a city

A normal distribution is used to determine gross variation in temperature from one day to the next.

$$\delta T = N(0, 3)$$

Then, single exponential smoothing is used to smooth temperature to seasonal normal, to avoid inconsistent temperature.

The α parameter determines the importance of flattening.

- 1: the temperature is entirely influenced by the seasonal normal.
- 0 : the seasonal normal is ignored.

$$TCity = TCity + \alpha(seasonalNormal - TCity)$$

Finally, to remain consistent with other cities, temperature of a city (called after the station) is influenced by temperature of the other cities the day before. A city close to another must have a similar weather.

$$TCity = TCity + \beta(TInfluence - TCity)$$

Influence temperature of other cities is the sum of temperature of other cities multipled by proximity of the city to the station.

$$TInfluence = \sum_x Tx \times Ix$$

City influence is the ratio of its distance to the station to the sum of all distances

$$Ix = 1 - \frac{cityDistance}{totalDistance}$$

$$totalDistance = \sum_x Distance(x, city)$$

2 Predicting the day's athmospheric pressure of a city

A similar process is used to predict athmospheric pressure of a city.

For each city, three local extrema (morning, afternoon and evening) of athmospheric pressure are forecasted.

Gross variation is determined by a normal distribution.

$$\delta P = N(1,0)$$

Once again, single exponential smoothing is used to smooth pressure to expected pressures, to avoid disproportionate values.

$$PCity = PCity + \Gamma(PSeasonalNormal - PCity)$$

3 Predicting weather (cloudiness, rain)

Thanks to pressure, altitude and temperature of the city we can make an approximation of the coming weather.

First of all, we need to get the sea level pressure, called P_0.

$$P_0 = P(1 - \frac{0.0065h}{T + 0.0065h + 273.15})^{-5.257}$$

T is temperature of the city P is the athmospheric pressure h is the altitude of the city.

Once P_0 is obtained, we can obtain a "weather indicator" called Z, which takes as value :

- If pressure is falling : $Z = 130 - \frac{P_0}{81}$
- If pressure remains steady : $Z = 147 - \frac{5P_0}{376}$
- If pressure is rising : $Z = 179 - \frac{2P_0}{129}$

Then,

- if current season is winter, $Z = Z - 1$
- if current season is summer, $Z = Z + 1$

(Normaly, wind direction affects weather, but as wind is not yet implemented, this part is currently ignored)

Finally, we obtain a number between 1 and 32, corresponding to the coming weather.

This method will be replaced by the cloudiness and wind data that we will calculate ourselves.

4 Predicting cloudiness

To estimate cloudiness value around the map, a matrix is built representing areas of the map with its cloudiness value.

From waters source, water is evaporated and form clouds. Then, clouds are dispersed around the map, but thanks to athmospheric pressures, clouds can more easily follow one direction or another.

5 Predicting wind

Currently, an edge detection is computed on cloudiness matrix to estimate wind on the map, using the following kernel :

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

6 Hourly forecasts

Polynomial interpolation is used to determine temperature, pressure and humidity at a specific time thanks to min and max values.

7 Obtain the temperature in any coordinates on the map

It is possible to estimate temperature and other indicators at every point on the map thanks to inverse distance weighting.

$$u(x) = \frac{\sum_{k=0}^{N} w_k(x)^p u_k}{\sum_{k=0}^{N} w_k(x)^p}$$

Where :

$$w_k(x) = \frac{1}{d(x, x_k)}$$

- u(x) is the estimated temperature of point x.
- d(x,xk) is the distance between x and xk.
- uk is the temperature of the station k.
- p is a power parameter. A greater value of p gives more importance to closer values of x.

The temperature of a point is influenced by each station on the map, but the closer the station is, the greater the impact on estimated temperature, and vice versa.

Part II
Data

Currently, data used is only seasonal normal temperature for each day, for each city, and athmospheric pressure.

City is described by its coordinates, its altitude and its seasons.

Seasons are described by their date, and their average values of athomspheric pressure.

A report is described by its minimal temperature (TMin), maximal temperature (TMax), date and athmospheric pressure.

1 Data Format

```
SeasonDef,Season_Name,Begin,End,Min_Pressure,Max_Pressure,Winter_Season
Region,RegionName,MapSizeX,MapSizeY,MapPath
City,City name,X coordinate, Y coordinate,Altitude
AddSeason,Season_Name
Date,TMin,TMax
...
EndCity
EndRegion
```

Example :

```
SeasonDef,StandardTatooine_Winter,01/01,01/06,1000,1020,True
SeasonDef,StandardTatooine_Summer,01/06,31/21,1015,1035,False
Region,Tatooine,1900,1250,tatooine.png
City,Bestine,1174,611
AddSeason,StandardTatooine_Winter
AddSeason,StandardTatooine_Summer
01/06/0280,18,36
02/06/0280,18,37
03/06/0280,18,37
EndCity
City,Carnthout,1280,711
AddSeason,StandardTatooine_Winter
AddSeason,StandardTatooine_Summer
01/06/0280,17,34
02/06/0280,17,34
03/06/0280,17,35
EndCity
EndRegion
```

Bibliography

[1] J. Sujit Shankar Poornima Selvaraj, Pushpalatha Marudappa. Analysis of weather data using forecasting algorithms. 01 2019.