Prediction of Precipitation Rate for Countries

Supervisor: Greg Baker

Saeed Soltani 301246870, Bita Shadgar 301286799 12/11/15 CMPT 732

1. Problem Definition

Climate change has become one of the most serious problems today. Governments and world leaders have emphasized on tackling this issue and requested more and prompt action from people all over the globe. Climate change has caused many regions to suffer from severe drought and higher temperature, on the other hand, it has also caused devastating rainfalls and changes in the ecosystem [1].

Drinking water is the most essential compound for fauna around the world. This valuable compound has become less and less available with the sudden grow in human population, air pollution and agriculture. Increasing temperature which is due to climate change has caused a significant impact on our fresh water supply [2]. In fact, it is estimated that over 1 billion people over the world facing lack of water and this number grows to 2/3 of the world by 2025. European Union has placed poverty and water shortage as the most serious problem in 21st century [3]. Experts have proposed number of solutions such as education to change consumption and lifestyles, invent new water conversation technologies, recycle wastewater and etc. But, none of these propositions have helped to reduce the consumption [4].

One of the solutions that could help governments to reduce the effects of water shortage is to have some estimates about the precipitation rate of their country in the upcoming years. Average precipitation estimation could aid leaders to take necessary measures including building dams on regions with higher rainfall or place water rationing on locations where the annual rainfall will decrease in the near future and some in-advance preparation is required.

The goal of this project is to predict the annual rainfall for countries from across the globe. In order to achieve this, dataset from the last 25 years (1990-2015) has been collected from the national centers for environmental information. This datasets is called global historical climatology network and it has daily information about the temperature, rainfall and snow rate of 99444 stations around the world [5]. Since the dataset is very large (25 GB), need of big data tools is necessary. Therefore, this project make use of Spark framework as the basis of the big data engine and also python programming language for coding. We hope, from predicting the precipitation rate for the upcoming years, we could play a small part in helping the governments to take necessary measures to prevent their nations facing drought.

2. Dataset

GHCN-D is a dataset that contains daily observations over global land areas [5]. Like its monthly counterpart, GHCN-Daily is a composite of climate records from numerous sources that were merged together and subjected to a common suite of quality assurance reviews. The archive includes the following meteorological elements such as daily maximum and minimum temperatures, precipitation (i.e., rain, melted snow).

There is an annual directory contains an alternate form of the GHCN Daily dataset. The by_year files are updated daily to be in sync with updates to the GHCN Daily dataset. The yearly files are formatted so that every observation is represented by a single row with the fields such as: Station identifier (CCDDDDDDDDD; where CC is country code, and the rest is station number; such as AE000041196); Date (yyyymmdd; where yyyy=year; mm=month; and, dd=day); Observation type; Observation value and four more fields that we do not use.

Also, there is information about stations in ghcnd-stations.txt. This file contains information of 99444 stations from whole world. This file is formatted so that each row is representing one station with the fields: Station identifier (CCDDDDDDDDD; where CC is country code, and the rest is station number); Station latitude; Station longitude and four more fields that we do not use.

However, since we are trying to visualize average precipitation of each country, we need to know which stations belong to which countries. Therefore, we need another file i.e. ghcnd-countries.txt that provide country code of 218 countries. Each row of this file include fields: Country code (CC; country code such as AE) and country name.

3. Methodology

This project is based on Spark python, therefore most of the computation and analyzation is performed on this framework. Also the dataset that is being used is in the CSV format. Thus, spark CSV library is included in the program to enable reading CSV format files. In terms of data aggregation and computation, Spark SQLContext and PySpark are implemented in the project. Regression and machine learning algorithms are used which are based on the Spark Mllib library. Figure 1 represents different parts of our project in a block diagram. The rest of this repot explains each part in more details.

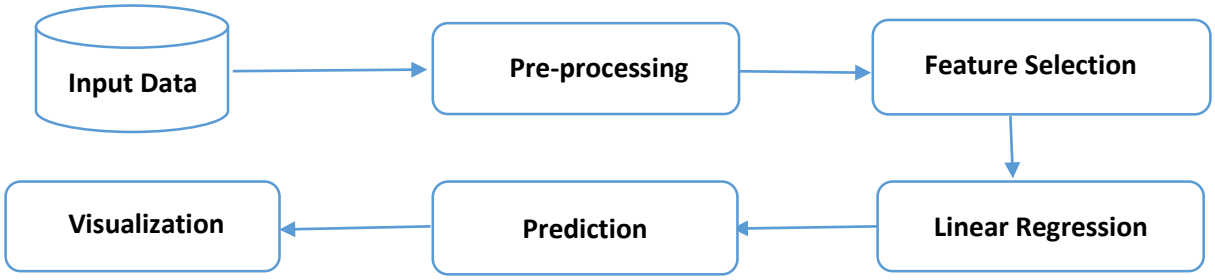


Figure 1: Project block diagram.

3.1 Preprocessing

GHCN dataset contained a lot of information, but we are interested only in the precipitation value of each station on a daily basis. Therefore, station, date and precipitation value of records that report the precipitation of a given station are extracted from the dataset and stored in database.

In order to calculate average precipitation of each station in a year, it is necessary to extract value of year from date column. Then an aggregation function in SQL is used to compute the annual average precipitation for each station. This task is executed through the SQLConetxt library. Table 'prcpMean' in database contains yearly average precipitation of each station which has a schema like:

prcpMean =(station-id, year, avg_prcp)

We have also applied more prepressing in order to provide the suitable data for visualization part which would be explained later in this report.

3.2 Feature Extraction

To train our model, there is a need to have more features to be able to predict more accurate. Geographical location of each station can be meaningful in this regard. Therefore, we extract latitude and longitude of each station from dataset, and join to the prcpMean. In order to prevent out of memory exception in SparkSQL, the SQL join has been achieved via Rdds in PySpark. Now we have a dataset ready to train which includes avg_prcp, station-id, year, latitude, longitude fields. For sake of better efficiency and memory usage, we have decided to store our final clean data in Parquet format which is a column-based type of storage

3.3 Linear Regression

Since we need a value as the result of our prediction, we have to use a regression model for our data. Indeed, we have huge amount of data to be trained and modeled, therefore we decided to use linear

regression with stochastic gradient decent which also exists in Mllib library of spark, named LinearRegressionWithSGD. This library provides a method to train the model which runs on train data. Train data has to have a specific format of LabeledPoint. LabeledPoint adds to attributes; one is label and the other is features. Based on this training model our data has been parsed and formatted, so that avg_prcp is considered as label and year, latitude and longitude are considered as features.

In order to estimate error rate of our model, hold out method is used. In other words, all parsed data has been divided in two parts i.e. train data and test data randomly. Train data is used to create the model and test data is applied to calculate minimum squared error.

There are some other parameters in train method which help to train the model more accurately. The general format of train methods is like:

train (*data*, *iterations=100*, *step=1.0*, *miniBatchFraction=1.0*, *initialWeights=None*, *regParam=0.0*, *regType=None*, *intercept=False*, *validateData=True*)

where **data** is training data, an RDD of LabeledPoint; **iterations** is used for number of iterations (default: 100); **step** is step parameter used in SGD (default: 1.0); **miniBatchFraction** is fraction of data to be used for each SGD iteration (default: 1.0); **initialWeights** is the initial weights (default: None); **regParam** is the regularized parameter (default: 0.0); **regType** is used to represent the type of regularizer used for training our model, and its value can be "l1" for using L1 regularization (lasso), "l2" for using L2 regularization (ridge), and None for no regularization (default: None); **intercept** is a Boolean parameter which indicates the use or not of the augmented representation for training data (i.e. whether bias features are activated or not, default: False); and **validateData** is a Boolean parameter which indicates if the algorithm should validate data before training (default: True).

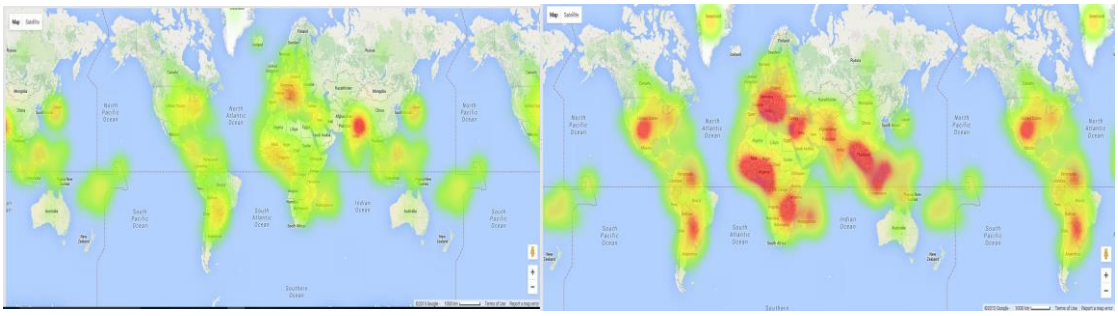
In order to reuse our model for different experiments, it is necessary to save the model, and for each experiment load the model. However, the Mllib library does not provide save and load methods for LinearRegressionWithSGD in python. Therefore, we have used java libraries to implement save and load methods. Furthermore, the Mllib linear regression model in Spark only learns from the values that are between 0 and 3, therefore, data rescaling and normalization is performed to scale the values between 0 and 3. This procedure is performed by finding the minimum and maximum values for each column and execute the normalization based on those numbers. These numbers are saved in a file called parameter for rescaling the predicted values in the prediction section.

3.4 Prediction

This module is responsible to load model and based on a query including our trained features i.e. year, latitude and longitude, predicts the precipitation rate of all the stations on the given year. Therefore, we have prepared all queries we would like to ask from our model in form of Labeled Points, but in this case the label is the station-id and the features are the same as trained mode. Again, scaling data and rescaling results are needed. Predict method runs on the model and extracts the results: model.predict *(features)*. The final results include station-id and the predicted value of precipitation for the given year.

3.5 Visualization

For visualization purposes Google Fusion [6] tables is used as a part of the visualization. Fusion table is a new API that Google introduced for editing CSV files and spreadsheets. This API also provides heat maps and feature information maps, if user's database includes any geographical information such as country, city or latitudes and longitudes. Figure 2, shows the generated heat map that Google Fusion tables have produced for 2020.The regions in the red sections, are the ones that face drought.



1990 2015

Figure 2: Heat map belong to year 1990 and 2015.

4. Problems

One of our main obstacles was due to tuning parameters of linear regression. In fact, to test our program in practice, we have reserved data of 2015 as final test, and excluded this year's data from training and modeling. It took us substantial amount of time to realize what is the best parameter; there were some cases that even though we had very little MSE on test data, but we had not the appropriate result for year 2015 as final test. Especially we learned that all data has to be scaled in [0, 3], otherwise the Mlib library would not learn from the imported parameters.

Indeed, we have used Google Fusion Tables to visualize the results, and this API has a limitation of representing 1000 locations on a map [7], while we have 99444 stations from all over the world to represent. Therefore, we solved this problem by grouping our data based on stations in each country and calculating the average for each country. We managed to overcome this issue by getting the code of each station and join this with another file that we later found from GHCN ftp directory.

5. Results

The results that we have received shows many interesting patterns. We expected that with growing danger of climate change, there would be a consistent decrease in the annual rain precipitation. However, after comparing the results of a few countries, we were proved wrong. As you can see in the Figure 3, Ethiopia has had a very inconsistent annual precipitation compared to Canada where every year the average has been very consistent. After some research, we realized this is due to a temperature anomaly called El Niño. El Niño is a climate cycle in the Pacific Ocean with a global impact on weather patterns [6]. This cycle begins from western tropical Pacific Ocean and usually ends near Indonesia and the Philippines [6]. El Niño has caused catastrophic effects on the climate and more importantly on the precipitation rate of the most regions in the world. The effects have been more severe in African regions. Studies shows that El Niño has caused one of the most severe drought periods in east and southern Africa and unfortunately lives of millions of peoples in this region are at risk from hunger, disease and water shortages [7]. This figure includes year 2020 which was predicted by our program.

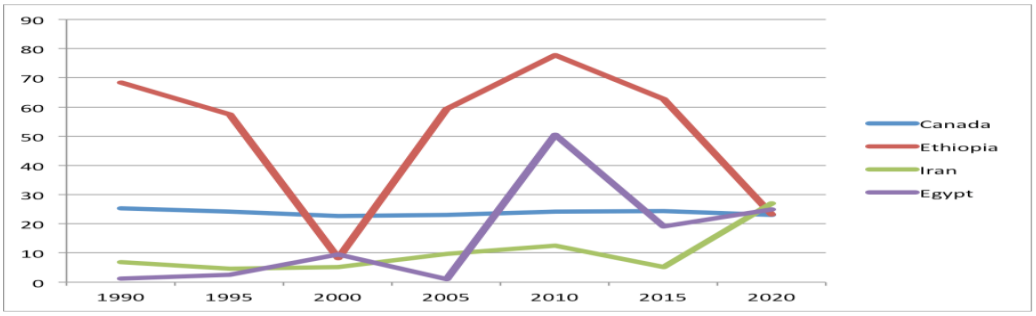


Figure 3. Flowchart of rain Precipitation

6. Summary

The dataset were downloaded from GHCN-D website through ftp request. Each year were separated by one csv files. Downloading this dataset were free to public. The dataset contained huge amount of information, therefore we extracted the necessary information from dataset and store it as Parquet files for memory efficiency. Our goal for this project was to have a small part in minimizing the damages of drought and climate change in the future, based on the error rate and numbers that our program produced, we believe these numbers could give researchers some ideas about the future of precipitation rate in the world. Integrating data and machine learning algorithms were the most challenging part of this project, Spark Mlib is a new library and it cannot provide all the required tools for implementing machine learning algorithms. Therefore, some steps such as scaling and rescaling the data, finding the optimized parameters and loading and saving the model were successfully applied to the project after multiple number of tests, debugging and researching. Our program managed to completely produce the results in about six minutes in the cluster without using the maximum power of the cluster. Given the huge amount of data and processing that were involved in this project, we believe the processed time was acceptable. The produced data is in the comma delimited format with name of the countries and their predicted precipitation rate in each line, hence, reading the data for users is very easy and they could view the data with any simple text editors, an output of the project is available in the Gitlab repository. For visualization, Google Fusion Tables were used, this technology enables you to view the data on the world map as a heat map or feature map and it is very easy to use. We realized presenting the data on the map would give a better understating to users about the generated output. Due to lack of saving and loading model in Spark Python, we added the Spark Java library to the project for this purpose. This model saves the model as a file on memory and could load it again when the path to the model is given. Technologies such as Spark Mllib for Regression, SQLContext for analyszing the data, Spark CSV for reading the files and PsySpark were part of this project. Overall, given that our ETL and cleaning dataset were not very difficult we believe 18/20 is a fair mark.

References

[1] Nature , “Nature Conservancy,” 2015. [Online]. Available: http://www.nature.org/ourinitiatives /urgentissues/global-warming-climate-change/threats-impacts. [Accessed 11 12 2015].

[2] Grace communications Foundation, [Online]. Available: http://www.gracelinks.org/2380/the-impact-of-climate-change-on-water-resources. [Accessed 11 12 2015].

[3] N. Jardine, "Business Insider," 7 October 2011. [Online]. Available: http://www.businessinsider.com/ the-10-biggest-problems-in-the-world-according-to-the-eu-2011-10?op=1. [Accessed 11 12 2015].

[4] "Experts name the Top 19 solutions to the Global Freshwater Crisis," Circle Of Blue, 24 May 2010. [Online]. Available: http://www.circleofblue.org/waternews/2010/world/experts-name-the-top-19-solutions-to-the-global-freshwater-crisis/. [Accessed 11 12 2015].

[5] G. H. C. Network, "Global Historical Climatology Network," National Centers for environmental information, [Online]. Available: https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/global-historical-climatology-network-ghcn. [Accessed 11 11 2015].

[7] L. Science, "What is El Nino?," Live Science, 20 August 2015. [Online]. Available: http://www.livescience.com/3650-el-nino.html. [Accessed 2015 12 12].

[6] BBC, "El Nino threatens 'millions in east and southern Africa'," BBC, 10 November 2015. [Online]. Available: http://www.bbc.com/news/uk-34779447. [Accessed 12 12 2015].

[8] Google, "About Fusion Tables," Google, 1 1 2014. [Online]. Available: https://support.google.com/ fusiontables/answer/2571232?hl=en. [Accessed 12 12 2015].