Analyzing Weather Data to Predict Extreme Weather (CSE 603)

Under the Guidance of: Prof. Vipin Chaudhary

Rakesh Baingolkar
Graduate Student (Computer Science)
University at Buffalo
Amherst, NY, USA
rbaingol@buffalo.edu

Manoj Mariappan
Graduate Student (Computer Science)
University at Buffalo
Amherst, NY, USA
manojmar@buffalo.edu

***Abstract*— Weather warnings are important as they give us time to protect life and property. Forecasts based on temperature and precipitation are important to agriculture. Forecasts are used by utility companies to estimate the demand over coming days. Outdoor sports are affected by heavy rain, snow etc. Last but not the least the airline companies plan the flights according to the weather forecasts. We plan to predict the extreme weather using the weather history of that particular location. We believe this will help the people in general to stay indoors whenever there is a weather alert. This will help them to stay safe in case of extreme weather and thus protect the loss of life and property. Our basic idea is to preprocess the available data to convert it into a form required to use machine learning algorithm and after running the algorithm we determine whether the weather is extreme. Basically we train a model using classification algorithm called Logistic Regression and eventually test the model on the testing set and find the accuracy of our model. The generated model will be used to predict the future weather extremes.**

***Index Terms*—Weather, forecasting, extreme, weather, Buffalo.**

I. INTRODUCTION

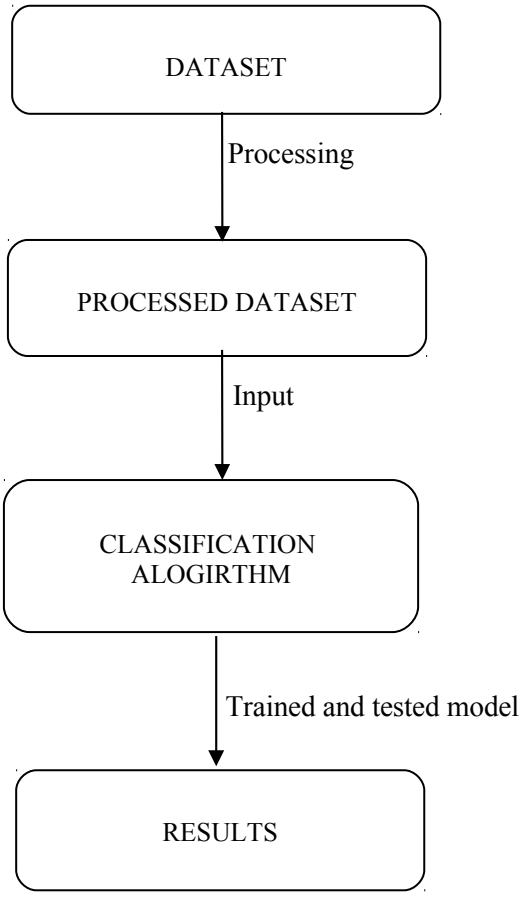
Weather forecasting is nothing but to predict the state of the atmosphere for a given location based on the current weather at that location. Ancient Chinese and Indian astronomers weather prediction techniques date back to 300BC. Formal weather prediction dates back to nineteenth century. There are various factors affecting the weather namely rain, soil temperature, wind speed, water temperature. General weather prediction done today is by sampling a wide amount of weather stations and use the satellite images to map out the positions of the large air masses surrounding the earth. Since the behavior of air masses is predictable as they interact in a specific way, meteorologists are able to predict the weather with some level of accuracy. Geography also has a hand in weather of a particular location i.e. weather prediction model generated at New York City cannot be used in Rio de Janeiro. Large bodies of water and mountains can affect the local weather. Wind patterns of a region also affect the local weather. Few facts about weather are that coastal areas have more moderate temperatures than inland areas, high altitude areas receive more precipitation than the low altitude areas and during the day in case of coastal areas he air moves inland during the day and away from the land during the night. Be it any place it is necessary to predict the weather as it affects life to a great extent. Anomalies in nature are rare but they do exist. Some blame the global warming to be the reason for many of the anomalies. Recent weather anomalies include the Lake effect snow in Buffalo which killed at least thirteen people and the second anomaly is the snowfall in Saudi Arabia. Tools used for weather prediction are barometer, anemometer, psychrometer etc.

Extreme weather means any weather that is unusual, severe or unconventional. It has hazardous effects on human life and property. Some reasons for extreme weather are carbon pollution but in many of the cases such as volcano the extreme weather is natural. One existing form of Weather prediction is the Numerical Weather Prediction. Numerical Weather prediction deals with taking the current observations and processing this data with computer models to forecast the future state of weather. Current weather observations serve as input to the numerical computer models and they use a method called data assimilation to predict the output of temperature, wind and various other parameters from the

ocean to the top of the atmosphere. We intend to predict extreme weather from this data set.

II. METHODOLOGY

Our methodology includes taking a data set, processing it to a form that can be used by a Classification algorithm. For classification algorithm we input this data set. The data set is divided into training and testing set and the algorithm is run on both the sets. The output of the algorithm is the trained model and the accuracy of that model.



II. A. DATASET

The data set that we used was from National Climatic Data Center. It is based in Asheville, North Carolina and it has the world's largest climate archive and it provided climatological services and data to every sector of United States. We collected weather data of Buffalo, NY from January 1st, 1950 to November 20, 2014. The size of the data was a little more than 1GB. The parameters of the data set included PRECIPITATION: Rain (mm), Snow (mm), SUNSHINE: Daily Percent (%), Daily Total (minutes), TEMPERATURE: Max Temp (Celsius), Min Temp (Celsius), WIND: Average Daily Wind Speed (m/s), Fastest 5 sec wind speed, WEATHER TYPE :

WT01: Heavy Fog
WT02: Freezing Fog
WT03: Thunder
WT04: Ice Pellets
WT05: Hailstorm
WT06: Glaze Ice
WT07: Volcanic Ash
WT08: Tornado
WT09: Drifting Snow

The above mentioned Weather Types are given in a binary format i.e. if the condition was present that day then the values of the column is 1 or else it is 0. This indicates that the weather was extreme that day. Thus the way we conclude that the weather was extreme that day is when any of the parameter is 1. In addition to the mentioned values we also had additional parameters such as fastest wind in 2 minutes etc. which we chose to ignore.

II. B. PROCESSING DATASET

For processing the data set we used python scripts. Processing involved two steps:

1. Extracting the concerned columns:

From the above mentioned parameters from the original data set we ignore non related parameters such as SUNSHINE, TEMPERATURE because we think they do not affect the possibility of extreme weather as compared to the other parameters like RAIN, SNOW and WIND. We extract RAIN, SNOW and WIND parameters along with the WEATHER TYPE parameters which indicate whether the climate was extreme. Further processing is done in the next step.

2. Adding 'Extreme' Column:

In this part first the values for which no data was available for some cases were given by -9999 in the dataset. These values were updated to 0 so that these values do not create an impact in our algorithm while classifying. Next we add an 'Extreme' column to the data set whose value is 0 if all the parameters in Weather Type are 0 and its value is 1 if any of the weather type parameter is 1 i.e. the weather was extreme that day.

So finally we have a dataset which has the following columns:

RAIN	SNOW	WIND	EXTREME
40	32	40	1
0	20	15	0

II. C. SELECTION OF ALGORITHM

First we decided of using Mongo DB and K means Clustering for our project. But after the processing stage we considered the form of the resulting data sets. After going through the videos of Andrew NG from Coursera we decided that our problem fits in the Classification Section of Machine Learning. The course documents were detailed for using Apache Mahout and thus we decided to use Apache Mahout instead of Mongo DB. Classification Algorithm is used for deciding if an email is spam/not spam depending on the parameters such as source, keywords used etc. It is also used for predicting whether a tumor is malignant or benign depending on the parameters such as tumor size. In our case we decided whether the weather is going to be extreme or not depending on the parameters RAIN, SNOW and WIND. The parameter which is decided is known as the target variable and the parameters which help in deciding the target variable are known as predictor variables. Classification in Apache Mahout can be done in various ways. It consists of Naïve Bayes, Logistic Regression and Random Forest.

III. IMPLEMENTATION

The primary task of implementing machine learning to a large set of data is deciding which approach of machine learning is to be used. The two main approaches are classification and clustering. Classification analyzes the input data set and builds a model on top of it. This model understands the dataset and based on the algorithm used by the model analyzes the pattern in which the input leads to a target and learns from it. Clustering on the other hand create clusters and assigns each input data to a cluster. The data that was input to the model also affects the cluster and thus decides the future flow of decisions.

The decision of which approach to select depends mainly upon the dataset we have. Clustering is primarily used to understand if the input dataset forms clusters to group themselves. Classification is useful when we have predefined classes and based on values of input we decide which class it belongs to. The primary purpose of our project was to decide if there would be an extremely climatic condition based on the weather conditions on that day. The three weather conditions in our dataset were continuous values denoting weather, wind and snow. Clustering becomes a little complicated for our dataset as it would create clusters and in the end we would decide which cluster represent extreme condition and which does not. Classification would be a better choice owing to its nature of determining if a given input value of weather, wind and snow would decide if it belongs to the class 'Extreme Weather' or 'Not an Extreme Weather'. As mentioned earlier, our dataset was obtained from the source and boiled down to input variables and a target variable which has the value 0 or 1; 0 represents non-extreme weather and 1 represents extreme weather. Thus, the classifier trains its model to determine on the basis of input if the output value is 0 or 1.

III. A. LOGISTIC REGRESSION

We further had many options under classification such as Naïve Bayes, Hidden Markov Models, Random Forest, etc. but arrived at the decision to use Logistic Regression. The other mentioned models rely on discrete variables and based on different possible values make a probabilistic decision on which class the current input belongs to. Since, in our case we have continuous values of weather conditions hence the decision to choose Logistic regression.

Stochastic Gradient Descent is an approach to discriminative learning linear classifiers like Logistic Regression. Our Model uses the logistic regression function to develop a model of stochastic gradient descent. Logistic regression function uses the logistic function to build dependency between the input and output variables. The logistic function is as follows:

$$\frac{1}{1+e^{-x}}$$

The range of the function lies between 0 and 1. The model in concern uses this function to handle multiple input values. The value of x in this case is

$$x = \beta_c + \beta_0 x_0 + \cdots + \beta_n x_n$$

$x_0, x_1, \ldots, x_n$ are values of input variables which in our case is wind, rain and snow. The βs are called regression coefficients. They help decide if an input variable belongs to class 0 or class 1. The model decides the value of regression coefficients through learning i.e. by analyzing the input values and their corresponding targets.

III. B. SPLITTING THE DATA

Before proceeding to perform analysis of data followed by learning by the model the data in hand which is the csv file obtained after pre-processing of data is to be read and handled. But, we also need to split the data so that part of the data is used for training the model and the rest is used for testing if our model functions properly. We read the csv file and read each input and load it into a vector of dense vector and also the target values into another vector. We select 75% of the dataset as the training set. The rest 25% enters the testing phase.

III. C. TRAINING THE DATA

In the training phase we create a model and pass each input and target value to the model sequentially. We created a model of online logistic regression. After testing the data we obtain that the results are as good as a random model. Thus, we upgraded to an Adaptive Logistic Regression model. The advantage of using a logistic regression model is that it creates multiple models under it and each of them are trained in a different way i.e. with different regression coefficients. There are as many as 20 different online logistic regression models with a different behavior and they run parallel in a distributed environment. Thus each of the models is trained in parallel with the single input. Although the online logistic regression model takes each input seriously, it is very fast in comparison to other machine learning algorithms. The adaptive logistic regression being built on top of online logistic regression is also fast and coupled with multiple models running in parallel shows an improved accuracy as well. Once the adaptive logistic regression model has been trained completely with the training set we select the best model from it. Once we obtain the best model we then move on to the next step i.e. the testing phase.

The parallelization in adaptive logistic regression is primarily in the training phase when multiple models are being trained at the same time. This makes use of the underlying Hadoop framework to handle various models. These models are maintained in a pool and has different learning rates. Once all the training dataset has been loaded to the model it can then be either serialized and stored for future use or used directly for prediction purposes.

III. D. TESTING THE DATA AND EVALUATING ACCURACY

The testing data which we had stored will now be utilized to evaluate the accuracy of the model. We run through each input value and pass it to classify function of the best model obtained from adaptive logistic regression. During the training phase the model had evaluated the regression coefficients which basically determines which input parameter has the highest effect on the output i.e. each value of climatic condition has different effects on the extreme weather condition. We compare the output we get with the actual target value. Thus we calculate the count of all testing data for which the output value matched the target value. This count divided by the count of training data gives the accuracy of the model.

III. E. FURTHER IMPROVISATION

An improvement over the algorithm was to have multiple passes of the input to the model. The input was shuffled and passed to the model in multiple passes or iterations. Each time the re-ordering helped the model learn in a different way. This coupled with different learning rates of the model helped obtain improved performance. We arrived at the decision to relate accuracy with the number of available cores. We again parallelize to train the data. The multiple threads train the different copies of shuffled input data. Thus we use the available cores to directly improve the accuracy. We tried integrating java implementations of OpenMP and MPI namely OpenMP/Java and MPIJava (open libraries available). Later, it was realized that basic multithreading in java would make use of the available nodes to serve the intention. We get the number of processors using the function availableProcessors() which returns the count of available cores and start as many as available threads. These threads do the work of only training a model. We dispatch all threads with their respective shuffled datasets and wait for all threads to complete. Once all the threads finish training the model we proceed to the testing phase. We observe that the improved training phase further improves the accuracy. As the model is trained again and again for different patterns of the same data it adapts.

IV. RESULTS

The application was written in Java. The implementation was as mentioned earlier. Initially tested on a small input size of 100 MB of csv data. After preprocessing of data, the csv file was passed as input to the Java application. The application was compiled as a jar file and passed to the Hadoop cluster with 3 nodes. The processing took approximately 5 minutes. Since, the data was small the accuracy obtained was low: between 60-70% only for multiple runs of the application.

```
SLURM Environment Variables:
Job ID = 3091570
Job Name = mahout_weather
Job Node List = d07n33s[01-02]
Number of Nodes = 3
Tasks per node = 2
CPUs per task =
/scratch/jobid = /scratch/3091570
Submit Host = k07n14
Submit Directory = /ifs/user/manojmar
```

```
ls: Cannot access .: No such file or directory.

run computation
training
Available Cores: 10
The accuracy of the logistic regression is: 66.64932
Time taken: 325 seconds
ls files in dfs
Copy output from HDFS to local directory
stop jobtracker (mapred)
stopping jobtracker
d07n33s02: stopping tasktracker
d07n33s01: stopping tasktracker
```

Following this, it was run for a file size of 1GB. This file consisted of weather data for New York State from 1950 till 2010. It ran for 1 hour 15 minutes approx. Although, due to large number of samples to learn from the accuracy boosted to 92-96%.

```
run computation
training
Available Cores: 10
The accuracy of the logistic regression is: 94.34085
Time taken: 435498 seconds
ls files in dfs
Copy output from HDFS to local directory
stop jobtracker (mapred)
stopping jobtracker
```

Thus, we inferred that accuracy improved as larger and larger samples were provided to Mahout.

V. OTHER APPROACHES

We initially tried to utilize the Weka library for machine learning which provides machine learning techniques to analyze given data in a sequential way. Following a lot of research in area we came across a parallel implementation of Weka which could analyze our data set in a parallel fashion. This was condemned later because it required to be installed on all the nodes which would be utilized for computation. Instead Mahout was preferred as it was available on all nodes and served the purpose of our project. Secondly, we also did some research on Hazelcast which is suitable for comparatively small amounts of data. It does face challenges with large amounts of data around TBs. But, for our case which had little amount of data of around 1GB Hazelcast would have sufficed. As it too required to be setup on all the clusters we switched to Mahout which served all the purpose and was available on all the clusters.

VI. CHALLENGES

The main challenge that we faced while doing the project is the selection of the data set to work on. The main reason for this was that all the data sets include common parameters like TEMPERATURE, RAIN, SNOW but not all the data sets contained extreme weather parameters like HEAVY FOG, THUNDER, and TORNADO. The National Climatic Data Center had these parameters so we decided to go with its data set. These parameters made it possible for us to use the classification algorithm on our data set. In any classification algorithm be it tumor being malignant of benign we have a history of data of all the parameters with the values of target variables. Other challenge that we faced was identifying the type of machine learning problem our problem fitted into. Machine Learning has a myriad of possibilities for a single problem to fit into. Some of the problems being Classification, Clustering, Regression, Recommenders. It took a while for us to figure out that ours was a Classification Problem. Other few challenges that we faced were selection what type of classification algorithm to use and last but not the least the processing phase of the dataset and convert it to the form used by the Logistic Regression phase. Apache Mahout expedited our implementation process.

VII. CONCLUSION

From the above experiments and observations we infer that a high accuracy is obtained for analyzing weather data to find out if there is a weather calamity expected to hit an area, provided we have large amounts of weather data. The accuracy is directly proportional to amount of data we have for the area under concern. Based on the weather history of the place we can predict extreme conditions and take appropriate actions such as alerting the residents.

VIII. FUTURE WORK

Although the model designed analyzes data to determine if an extreme condition is expected to occur an even better and useful thing would be to determine which particular condition is going to occur. With even better information on weather conditions we would have information on which particular extreme condition occurred such as a snow storm, hail storm, hurricane, etc. This leads to multiple target variables in contrast to our model which had binary target. This could be implemented using multinomial logistic regression. It is currently not available on Mahout but could be implemented through other languages such as R and apache spark.

IX. REFERENCES

[1]http://computing.ornl.gov/workshops/scidac2010/presentations/v_kumar.pdf

[2] Some tips for predicting weather http://www.ussartf.org/predicting_weather.htm

[3] Data Set: http://www.ncdc.noaa.gov/

[4] Apache Mahout: http://mahout.apache.org/

[5] Mongo DB: http://www.mongodb.org/

[6] WEKA Parallel: http://weka-parallel.sourceforge.net/report.pdf

[7] For extracting scripts: www.stackoverflow.com

[8] Idea on performing logistic regression: http://blog.trifork.com/2014/02/04/an-introduction-to-mahouts-logistic-regression-sgd-classifier/