Sentimental Analysis using Spark Streaming

Manish Kumar*
Kamyar Setayesh Ghajar*
m56kumar@uwaterloo.ca
k.ghajar@uwaterloo.ca
University Of Waterloo
Waterloo, Ontario, Canada

Abstract

Many "big data" applications must act on data in real time. Running these applications at ever-larger scales requires parallel platforms that automatically handle faults and stragglers. Unfortunately, current distributed stream processing models provide fault recovery in an expensive manner, requiring hot replication or long recovery times, and do not handle stragglers. Discretized streams (D-Streams) overcomes these challenges. D-Streams enable a parallel recovery mechanism that improves efficiency over traditional replication and backup schemes, and tolerates stragglers. In our project we try to solve the problem of predicting the US 2020 Presidential election. We make use of twitter data to fetch the tweets in real time. Based on the tweets, we perform sentimental analysis on the data. As, D-Streams can easily be composed with batch and interactive query models like MapReduce and other transformations can be easily applied, it becomes easier.

Keywords: Big Data,Spark Streaming, D-streams, Map Reduce,Sentimental Analysis

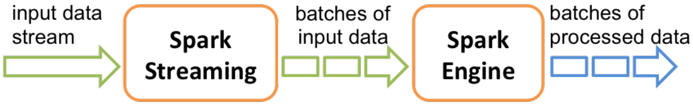
1 Introduction

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Spark streaming is used for receiving a stream of data in real time. We assume that stream of data arrives as windows/chunks of data and time defines the order of stream elements. We fetch the tweets of US citizens as it will be unbiased analysis and we try to gather insights from their tweets by performing sentimental analysis. Sentimental analysis is a Natural Language Processing Algorithm used to interpret and classify emotions in subjective data.

2 Discretized Streams (D-Streams)

D-Streams avoid the problems with traditional stream processing by structuring computations as a set of short, stateless, deterministic tasks instead of continuous, stateful operators. They then store the state in memory across tasks as fault-tolerant data structures (RDDs) that can be recomputed deterministically. Decomposing computations into short tasks exposes dependencies at a fine granularity and allows powerful recovery techniques like parallel recovery and speculation. Beyond fault tolerance, the D-Stream model gives other benefits, such as powerful unification with batch processing.

Figure 1. D-Streams



3 Technologies Used

3.1 Spark Streaming

Spark Streaming is an add on package to the core Spark API which is scalable, highthroughput and also fault-tolerant. It provides functionality of processing live data streams. For streaming Apache spark can have flume, HDFS, apache Kafka, twitter, kinesis data sources. This data can be then cleaned and structured in spark itself and used to do further processing.

3.2 Spark SQL

Spark core has an SQL extension which supports more optimization on datasets(RDD) and is in structured format to retrieve data using the SQL queries. Spark SQL provides most convenient way to perform several transitions on the data. Spark SQL uses dataframes for data manipulations

4 Problem Identification

The problem we solve in this project is to predict the winner of US Presidential Elections 2020 in a state. As predicting the overall winner is not a trivial case, if we can focus on a state at a time we can achieve fairly decent accuracy and then we can upscale it for all the 50 states. Having a Twitter developer account helped us in fetching the tweets of different users in real-time. We made sure that the tweets arrive from only that particular state(chosen at that instance) in our case, and also the place they are tweeting from must be the same state. We repeated this process for all the 50 states and took an average of all the 50 states to declare the overall winner. We also made sure that no redundant data is stored in our model by removing the tweets of a single user tweeted multiple times. Hence we removed discrepancies from the model and that helped us in achieving good results.

5 Data Description

Twitter is a social networking and microblogging service that allows users to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging service (quick and short messages), people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings. Following is a brief terminology associated with tweets. Emoticons: These are facial expressions pictorially represented using punctuation and letters; they express the user's mood. Target: Users of Twitter use the "@" symbol to refer to other users on the microblog [1]. Referring to other users in this manner automatically alerts them. Hashtags: Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets. We classify tweets into positive, negative and neutral labels. We make use of TextBlob library which is based on NLTK.

6 Data Pre-Processing

The collected data were noisy, missing useful info and inconsistent [1]. As, we receive the tweets in real time and on social media we don't have any objections on the tweet content, there are all kinds of tweets, some contains video,photos,emojis. In the sentimental analysis we are only interested in the textual data so we have to remove all other types of data.In the Data Pre-Processing, we have to check if there are empty values or inconsistency in the data. The data should be in a consistent state to be analyzed. To improve the efficiency of our analysis the data should be in a simple format. Data mining is done on this data so to get efficient results the data must be processed by removing redundancies. The data is made meaningful by deriving information from it like deriving location from the tweet, twitter id, user name of the tweet. We can derive the location of a tweet by looking at the latitude and longitude attribute of the tweet.

A schema of organized data which is received after preprocessing can be seen as follows:

```
DataFrame[polarity: bigint, subjectivity: string, tweet: string]
root
|– polarity: long (nullable = true)
|– subjectivity: string (nullable = true)
|– tweet: string (nullable = true)
```

7 Methodology

To know the sentiment of a textual data there are many algorithms which are based on NLTK [4]. In our project we have made use of one of those algorithms which is sentimental analysis.

7.1 Sentimental Analysis

Sentiment analysis, also refers as opinion mining, is a sub machine learning task where we want to determine which is the general sentiment of a given document. Using machine learning techniques and natural language processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive, neutral or negative. It is a really useful analysis since we could possibly determine the overall opinion about a selling objects, or predict stock markets for a given company like, if most people think positive about it, possibly its stock markets will increase, and so on [5]. Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...) but it is also why it is very interesting to working on.

7.2 About the Model

We have made use of TextBlob package for our model. TextBlob is a python library for processing textual data. So what happens is TextBlob applies sentiment analysis on each and every tweet and returns the polarity and subjectivity of that tweet. Polarity has three values -1,0,1 where -1 refers to the negative sentiment, 0 refers to the neutral sentiment and 1 refers to the positive sentiment [8]. We track the tweets based on keyword Trump and US Presidential Elections 2020,more keywords can be added to filter tweets but these two does the work. We create a streaming window of 5 seconds and for every 5 seconds batches of data arrives as a group and then its divided into chunks of batches(D-streams) which are stored as RDD's internally. We then apply sentimental analysis on each tweet and store it into Spark SQL Table as a tuple of (tweet,polarity,subjectivity). Where subjectivity refers to some personal feelings, views, or beliefs. We then calculate As Trump served the presidency in the last term, we expect to gather positive sentiments about him if his last term was appreciated by the people. If we got neutral sentiments or negative sentiments that means his tenure was not upto the mark and the chances are quite high that he might loose. We calculate the amount of positive sentiments and negative sentiments for each and every state and finally we take the average value to predict the overall winner.

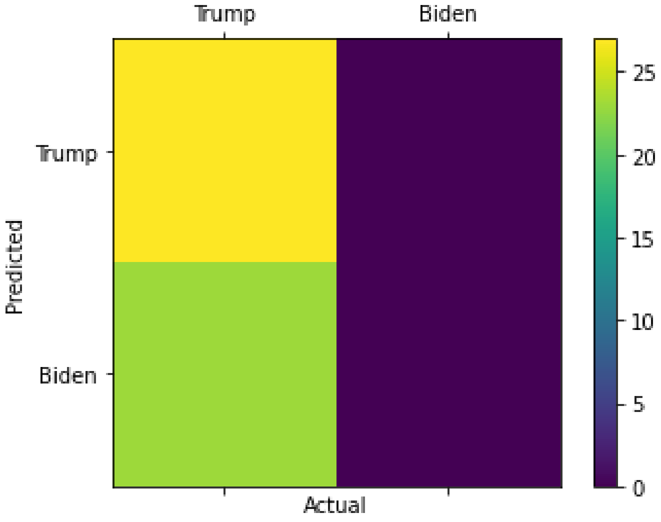
8 Results

Our model achieved an overall accuracy of 54%, though it seems less but predicting the winner of presidential election with a high accuracy is not trivial.

Table 1. Classification Report

	precision	recall	f1-score	support
0	0.54	1.00	0.70	27
1	0.00	0.00	0.00	23
		accuracy	0.54	50
macro avg	0.27	0.50	0.35	50
weighted avg	0.29	0.54	0.38	50

Figure 2. Confusion Matrix



There are several metrics proposed for computing and comparing the results of our experiments. Some of the most popular metrics include: Precision, Recall, Accuracy, F1-measure, True rate and False alarm rate (each of these metrics is calculated individually for each class and then averaged for the overall classifier performance.) As we can see our sentimental analysis model does a great job in predicting the winner. The False Negative and True Negative are more than 25, as a result the model does good job in identifying the opposite class but the type-II error is also large. As we can see from the classification report, the precision for class 0(Predicting Trump) as winner is 54 %. The F1-score was 0.70 that means 70% of our positive predictions were correct. Recall for class 0 was 1.00 which means the model was able to capture all the positive cases(which is the case where Trump triumphs). The model had a large type-II error which reduced the overall accuracy drastically.

9 Future Works & Conclusion

Sentimental analysis is already a hot-topic and the very idea of getting the sentiments of a text has been used by many companies to get a feedback from their customers. Many companies also look at twitter data to get the feedback. As millions of twitter users tweet regularly the data can arrive in real time and necessary changes has to be made in the model so that the data can be received in real time that enables scalable, high-throughput, fault-tolerant. Spark streaming enables handling the real-time data with high-throughput.

As, our data was limited to the tweets containing the names of the politicians that we defined as search terms. Therefore, we may have missed some replies belonging to a discussion thread because respondents do not necessarily repeat these names in every message. In their study of political discussion boards, Jansen and Koop (2005) have found that only 60% of all messages mentioned a political party by name[2]. However, since Twitter users are aware of the unstructured nature of microblogging communication and therefore include searchable keywords, so-called hashtags, in many messages (e.g., "Biden"), we believe the share of relevant replies to be small. In addition, parts of the information relayed through Twitter are embedded in links. Including these missing pieces of information may change our results regarding sentiment and equality of participation. Therefore, future research should try to capture the context of a particular statement more comprehensively either by following embedded links or by searching for replies to an author.

Improved accuracy and consistency in text mining techniques can help overcome some current problems faced in Sentiment analysis. Looking ahead, what we can see is a true social democracy that will be created using Sentiment analysis, where we can harness the wisdom of the crowd rather than a select few “experts”. A democracy where every opinion counts and every sentiment affects decision making.In the future we can make use of Apache flink which is much faster than spark streaming.

References

[1] A. I. Baqapuri, S. Saleh, M. U. Ilyas, M. M. Khan, A. M. Qamar, "Sentiment classification of tweets using hierarchical classification", 2016 IEEE International Conference on Communications (ICC), Malaysia, pp.1-7, 2016.

[2] Tumasjan A, Sprenger TO, Sandner PG, Welpe IM, “Predicting elections with twitter: What 140 characters reveal about political sentiment”, Icwsm Vol.10, Issue.1, pp.178-85, 2010.

[3] R.V. Patil, S.S. Sannakki, V.S. Rajpurohit, "A Survey on Classification of Liver Diseases using Image Processing and Data Mining Techniques", International Journal

of Computer Sciences and Engineering, Vol.5, Issue.3, pp.29-34, 2017.

[4] Madnani N, “Getting started on natural language processing with Python”, Crossroads, Vol.13, Issue.4, pp.5-9, 2007. [5]. B. Pang, L. Lee, S. Vanity Nathan, “Sentiment classification using machine learning techniques”, In Proceedings of the ACL02 Conference on Empirical Methods in Natural Language Processing, Vol.5, Issue.4, pp.79-86, 2002.

[5] B. Pang, L. Lee, S. Vanity Nathan, “Sentiment classification using machine learning techniques”, In Proceedings of the ACL02 Conference on Empirical Methods in Natural Language Processing, Vol.5, Issue.4, pp.79-86, 2002.

[6] Y. Yamamoto, T. Kumamoto, A. Nadamoto, “Role of emotions for multidimensional sentiment analysis of twitter”, In Proceedings of the 16th International Conference on Information Integration and Web-based Applications, USA,pp.107-115, 2014.

[7] V. N. Khuc, C. Shivved, R. Namath, and J. Ramayana., ”Towards building large-scale distributed systems for twitter sentiment analysis”, .In Pro-ceedings of the 27th Annual ACM Symposium on Applied Computing, pages 459-464, 2012. [8]. Dean J, Ghemawat S. “MapReduce: simplified data processing on large clusters”, Communications of the ACM, Vol.51, Issue.1, pp.107-113, 2008.

[8] H. Yuh, V. Huitzilopitchli, “Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences”, In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, USA, pp.129-136, 2003.

[9] L. Zhuang, F. Jing, X.-Y. Zhu, “Movie review mining and summa-rization”, In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, USA, pp.43-50, 2006.

[9] T. Wilson, J. Wienie, P. Ho Mann, “Recognizing contextual polarity in phraselevel sentiment analysis”, In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, USA, pp.347-354, 2005.