Project Proposal

The disagreeable frogs

1155129475 LUO Xuqi
1155129451 KAI Weiheng
1155132004 LI Ruochong
1155132913 PENG Du

Sentiment Analysis about NASDAQ 100 on Twitter

Abstract

Public voice is crucial for a company. With millions of active users, Twitter is simple, fast and convenient place for a person to post or comment a tweet to share his or her views about everything. In this paper, an original sentiment analysis algorithm, named sentiment K-means++, analysed public opinion for NASDAQ 100 companies. All the experiments are done on real-world data set collected from Twitter and over 2 million records.

Introduction

The response of market for a product or a company is crucial for many respectives. For example, for manufacture, the market acceptance is an extremely crucial indicator for products R&D, marketing strategy and manufacturing planning. Besides, the financial market regards the positive common of a product or a company as a bull new for its stock price, where tranmanous amount of profit can be made, if investors can capture the correct trend instantly. However, companies use questionnaires or phone calls to study public opinions which takes a lot of time and effort, but it is short of representativity. Thus, a system, check the comment of the public in real-time, solves such problem more effective and more efficient.

Conventionally, artificial neural network approaches require a significant amount of labeled data to train a classificator. From another hand side, clustering method is rarely used in the sentiment analysis task because vectorized content can be clustered not only by sentiment information, but by semantic or other dimensions. Additionally, with users generating the new twitts rapidly, clustering speed is required by such real-time sentiment analysis task.

Thus, we proposed a semi-supervised clustering algorithm, sentiment K-means++, for this task. Particularly, we lock on the clustering dimension on the sentiment information by arbitrarily initializing the centroid as the centroid of very positive and very negative class from a labeled dataset. Additionally, to improve the real-time processing ability, sentiment K-means++ integrated MapReduce method to find the new centroids in every step.

Related work

Sentiment analysis

Sentiment analysis is a job composed by data retrieving, preprocessing and clustering or classification [1][2]. Usually, the text based information is converted into vocabulary, but relations among words are hard to be captured[3]. Thus, converting text into vectors is more popular[3]. Within the converting process, pretrained model, such as word2vec, glove and BERT dominate this area. After the conversion, vectors from words with similar meaning should have similar distance. Then there are a number of methods to conclude the sentiment of a single record.

Conventionally, classification is the most popular approach to achieve a relatively high prediction accuracy [2]. For example, statistical classificator, such as bayes classificator, SVM (support vector machine) and KNN (k nearest neighbors), determinate the class information of an upcoming record according to predefined class information, class labels or the number of classes. Later on, artificial neural network based techniques thrived in recent years [3]. For instance, both RNN (Recurrent Neural Network) and CNN (Convolutional Neural network) approaches claimed a very positive result. However, such methods require a considerable amount of labeled data and a tremendous training time.

Moreover, clustering is also a method to do some predictions. For instance, K-means, DBSCAN, are well-known algorithms that can complete tasks. There is no need to pre-labeled the datasets, so it can produce the result without human interference. However, unsupervised method achieve weaker accuracy than those supervised methods.

In some cases of sentimental analysis, classification and clustering method are used as a combination to do identification and prediction on hotspots events. Normally, this kind of method starts at a calculation of the emotional polarity of a piece of text and to obtain a sentimental score of the text. The result of this step usually are vectors that contain some features to represent the corresponding piece of text.

After that, clustering could be used to group the data into several groups. For example, K-means can output the result which contains clusters and also the theoretical center of each cluster. Classification can then use the result to do training. The output of clustering as the supervised learning output, in order to train a classifier, such as SVM. Finally, the trained classifier can be used to do the prediction, and the result can be tested by comparing to the result of the clustering method to see the accuracy [4].

Distributed-based classification or clustering

Classification like KNN, the larger set of data, the more accurate the result could be. However, handling large sets of data is difficult to process. Therefore, many attempts on combining the classification or clustering methods and big data techniques have been conducted in these years. The pre-process steps are more or less the same, which format the input of training data into <identity, attribute1, attribute2, attribute3, ...>. Then for the input of testing data, it should be <ID, attribute1, attribute2, attribute3, ...>. While in the mapper and reducer, procedures could be various. For example, in one case, KNN was customized, in which K-value has been changed to the threshold of the distance between the testing data and the training data. This K-value could also be user-provided as well. Therefore, after computing the distance, the output would be <ID, identity>. The identity came from those training data which had a lower distance value with the testing data than the threshold K-value. At last, the reducer would count the identity for each testing data according to the key, ID of each testing data. The one appears most often would be the identity of that testing data [5].

There are many approaches that implement the classification or clustering based on Hadoop, Spark or any other Big data techniques, like naive Bayes, SVM, etc. Moreover, MapReduce could also improve the feature selection in classification [6]. These kinds of attempts can inspire us to do the implementation.

Method

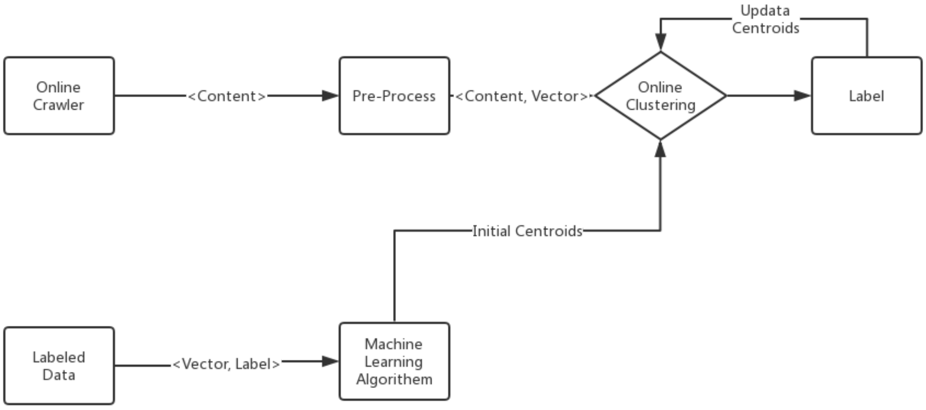

Figure 1. Data Workflow

In this task, a semi-supervised online k-mean clustering algorithm converts the content of twitters into sentiment vector. Particularly, a sentiment vector including both the polarity and the intensity. For example, “I like Amazon Very much” indicates a positive mood, which is polarity for positive, with strong intensity. The following sections explain the whole data pipeline.

In the beginning, a crawler collects tweets related to over 100 companies in the US stock market. Due to Twitter’s anti-crawl mechanism, Twitter’s official API will be used to retrieve the tweets, for example, tweets that are tagged with “AMZN” for Amazon. Everything in the content of twitters are preserved, including Emoticons and Emojis. Meanwhile, the sentiment 140 data set, where over 1.6 million twitters record labeled with sentiment information positive or negative, is collected for semi-supervised learning.

During the data-cleaning phase, tweet-preprocessor, which is a processing library for tweet data written in Python, is used. This library is able to clean, parse or tokenize the tweets[7]. Additional information, such as time, user ID, and other descriptive information will be assigned as attributes to each Tweet. Then, a per-trained machine learning model, doc2vector, transforms the text, from both crawler and sentiment 140, into vector, where the word or document with similar meaning are expected to have close distance. The data attributes and correspondent descriptions are summarized in the table below:

No.	Attributes	Description
1	id	Tweet’s unique id
2	created_at	Tweet’s date and time
3	source	Tweet’s source (published using

		web/Android/iPhone)
4	text	Tweet’s content
5	lang	Tweet’s language
6	favorite_count	Number of favorites per tweet
7	retweet_count	Number of retweets
8	original_author	Tweet’s author’s profile username
10	possibly_sensitive	Sensitivity of the message (Boolean: true/false)
10	Hashtags	All the hashtags contained in the Tweet
11	User_mentions	Other people mentioned in the Tweet
12	Place	User’s location
13	Place_coord_boundaries	Coordinate of the weet's location(If applicable)

Table.1 Data Attributes and Descriptions

Following, the special K-mean algorithm is used to construct an online clustering machine. There are two questions to be addressed in this stage. First, word or document vectors exist in an extremely high dimensional space. In other words, not only sentiment information is one of the dimensions, but semantic information or others can be described in such vectors as well. Thus, a simple random initialized clustering algorithm, such as naive K-mean, may associate data points into one cluster according to non-sentiment information which is not acceptable for our final outcome. Similar failure may happen, if the simple random initialization online K-mean clustering is implemented after the crawler from scratch. Secondly, features of different data sets, such as mean and distribution, may be variant. If the absolute centroid of every class of sentiment140 dataset is assigned for the classification task, the accuracy is expected to be not reliable due to the overfitting problem. Therefore, the reliability of various dataset should be considered as well.

Similar to K-means++ algorithm [8], we select initial centroids to reflect the sentiment information, rather than random initializing two centroids. For instance, as the polarity from 1-5 in the sentiment 140 represented the altitude level from negative to positive, the centroids of the most positive and negative classes are selected as the initial centroid. Sentiment information, for both positive and negative ones, has a higher probability to be captured by the clustering algorithm, because the most positive and negative points have the greatest intensity, but total opposite polarity in the desired dimension. Then, centroids will literate for times on the sentiment 140 datasets to achieve better performance on the data set from the crawler. In the gradient stage, even calculating distance between every single point and centroid cost a lot of computing power, MapReduce helps to accelerate the gradient speed. For instance, mappers calculate the distance for every class unknown point to pop out <class, vector>. Then, reducers find out the new centroids for this step.

To cluster real-time tweets, the program holds two centroids from the last step. For every newly arrived tweet, the algorithm determines the belonging cluster according to the distance between two centroids to this point. For example, assumed d1 and d2 are distances from centroid c1 and c2 to a new point, if d1 is smaller than d2, new point will be clustered to the first group. Then, the program constantly updates the new centroid to fit new data point.

In the end, we shall visualize the result as pictures. For example, we will assign the polarity and the intensity for every single twitter for a product. The value is the coordination for plotting where boundaries are expected to show between classes. Additionally, we should compare the classes distribution with different products and the variation of classes distribution for the same product cross time. Finally, a formal conclusion should be made to answer the topic question. For example, “iPhone 11 is not the most popular iPhone in the history, but it is the most fashiona phone in the world for now”.

Proposed Timeline

10.01	Crawl data from twritter.com
10.15	Preprocess
11.01	Apply clustering model
11.15	Draw the result, visualization
11.30	Conclusion
12.02	Presentation

Reference

[1] X. Hu, J. Tang, H. Gao, and H. Liu, “Unsupervised sentiment analysis with emotional signals,” *Proceedings of the 22nd international conference on World Wide Web - WWW ’13*. 2013.

[2] S. Alhojely, “Sentiment Analysis and Opinion Mining: A Survey,” *International Journal of Computer Applications*, vol. 150, no. 6. pp. 22–25, 2016.

[3] M. Lan, Z. Zhang, Y. Lu, and J. Wu, “Three Convolutional Neural Network-based models for learning Sentiment Word Vectors towards sentiment analysis,” *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016.

[4] N. Li and D. Wu, "Using text mining and sentiment analysis for online forums hotspot detection and forecast", *Decision Support Systems,* vol. 48, no. 2, pp. 354-368, 2010. Available: 10.1016/j.dss.2009.09.003.

[5] Q. Ding and R. Boykin, "A framework for distributed nearest neighbor classification using Hadoop", *Journal of Computational Methods in Sciences and Engineering*, vol. 17, pp. S11-S19, 2017. Available: 10.3233/jcm-160676.

[6] D. Peralta, S. del Río, S. Ramírez-Gallego, I. Triguero, J. Benitez and F. Herrera, "Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach", *Mathematical Problems in Engineering*, vol. 2015, pp. 1-11, 2015. Available: 10.1155/2015/246139.

[7] Kalpa, Dilan. “Extracting Twitter Data, Pre-Processing and Sentiment Analysis Using Python 3.0.” *Medium. Towards Data Science*, April 3, 2019.

[8] Arthur, David and Sergei Vassilvitskii. “K-means++: the Advantages of Careful Seeding.” *SODA,* 2007.