Efficient stock subsequence matching in time series using Hadoop

Name: Lei Yang U06367963, Shuhan Liu U09073975

Abstract

Match the stock subsequence in time series efficiently is very important in Stock trend analyze section. This paper discussed how to use dynamic time warping algorithm to query the stock subsequence in time series. It also discussed how to use Hadoop and MapReduce techniques to execute large amount of matchings efficiently.

Introduction and Motivation

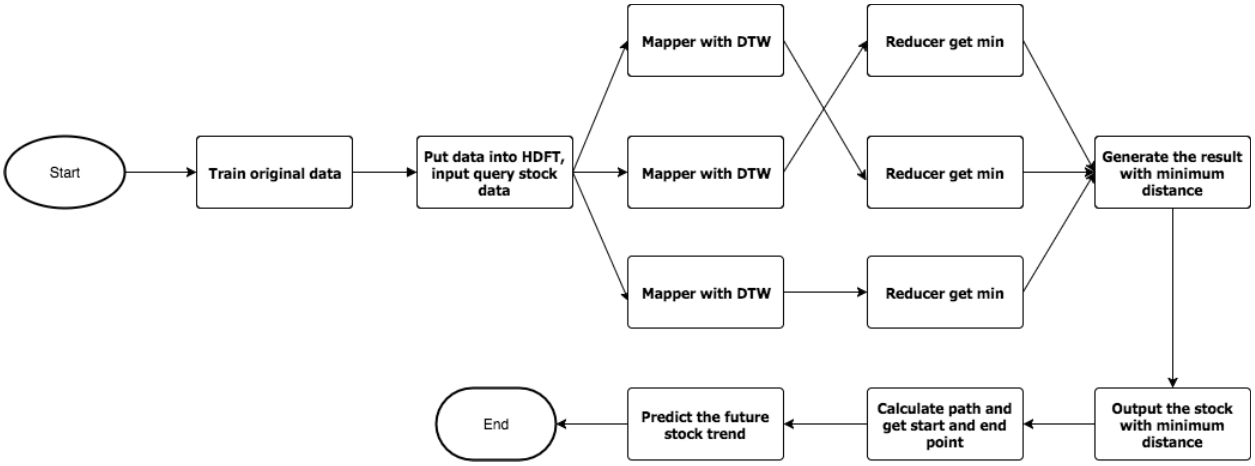
Find out the stocks with similar trend are very important in the section of stock trend analysis. Because two stocks with similar trends may have similar future trend. For example, if we want predict the future trend of stock A, we can take the stock A's past 3 months trend, and match the stock with most similar historical trend from the pool. Say we found stock B. Because we already know the historical trend of stock B, based on this, we can predict the future trend of stock A.

However, there are over 8,000 different stocks in United States, and most of them have over decades transaction history. It is very time-consuming to query the stock in time series without proper tool.

Problem Definition

1. Develop a program to query a given stock trend from another stock's historical data, calculate the distance of the time series between two stocks
2. Solve how to proceed problem 1 in very large amount efficiently. (say we have over 8,000 stock historical data with over decades length)

Approach and Method


Start
Train original data
Put data into HDFT, input query stock data
Mapper with DTW
Mapper with DTW
Mapper with DTW
Reducer get min
Reducer get min
Reducer get min
Generate the result with minimum distance
Output the stock with minimum distance
Calculate path and get start and end point
Predict the future stock trend
End

1. **Implement Dynamic Time Warping Algorithm**

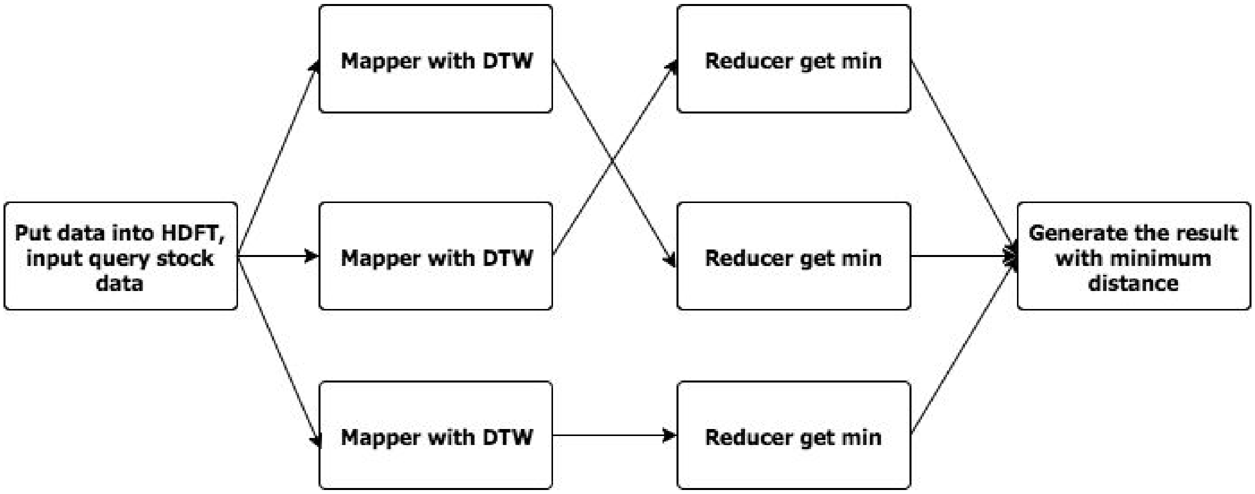
   Let source and target be one dimension double type matrix. Source is query stock time series, Target is target stock time series.
   Let i, j be the length of source and target array.

   I built a two dimensions distance matrix and first initialize the D0 matrix with the Euclidean distance between point source[i] and target[j]. D0 is a i * j matrix

   Then, based on the following minimum distance algorithm from 0 to i from source, to 0 to j from target
   D0[i + 1][j + 1] += Math.min(D0[i][j], Math.min(D0[i + 1][j], D0[i][j + 1]))
   D0[i][j] is the minimum overall distance matrix for the query, and D0[-1, -1] is the overall minimum distance.

   Return D0[-1, -1] / (i + j) and get the average minimum distance.

2. **Apply DTW to Hadoop and MapReduce**



Using Hadoop MapReduce, we can calculate the DTW very efficiently.

2.1. Use HDFS to manage the data

With HDFS, we can split the input file into blocks, each block has 128MB. And later each node it can be parallelly mapped and reduced

2.2. Map the data with DTW

We first implement the Dynamic time warping algorithm in Mapper, so that for each row of input, we can get the minimum distance between stock query and target stock.
e.g. the input row is: *apple: 48.485,45.858,45.633,47.855,47.198,46.532,...*
the row after the mapper is: *apple: 3*
Which means the distance between the query and stock apple is 3

2.3. Reduce the data and find minimum distance

After we mapper all the input data, now we have the distance result of each stock to query. Then compare the value of these distance and find the minimum, return the minimum distance among these stocks.

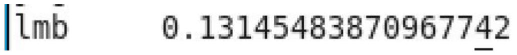
3. **Track the path and find the start point**

From step 2, we already know which stock has the minimum distance with the query. Now we can take a further step and find the range of the time series. It's easy to realized by add a track note in the cost matrix calculated in step one, and track back to get the start and end point, which is the range of time series.

Experiment & Results

We randomly pick a range of time series of stock from the input as the query:
14.793,14.813,14.96,14.911,14.695,14.557,15.383,14.842,14.881,14.646,14.616,14.754,14.793,14.823,14.901,14.842,14.911,14.744,14.754,14.852,14.685,14.911,14.734,14.646,14.695,14.626,14.615,14.724,14.646,14.41,14.44,14.528,14.714,14.744,14.665,14.685,14.449,14.164,14.184,14.164,14.066,14.007,13.958,14.095,14.252,14.38,14.4,14.302,14.223

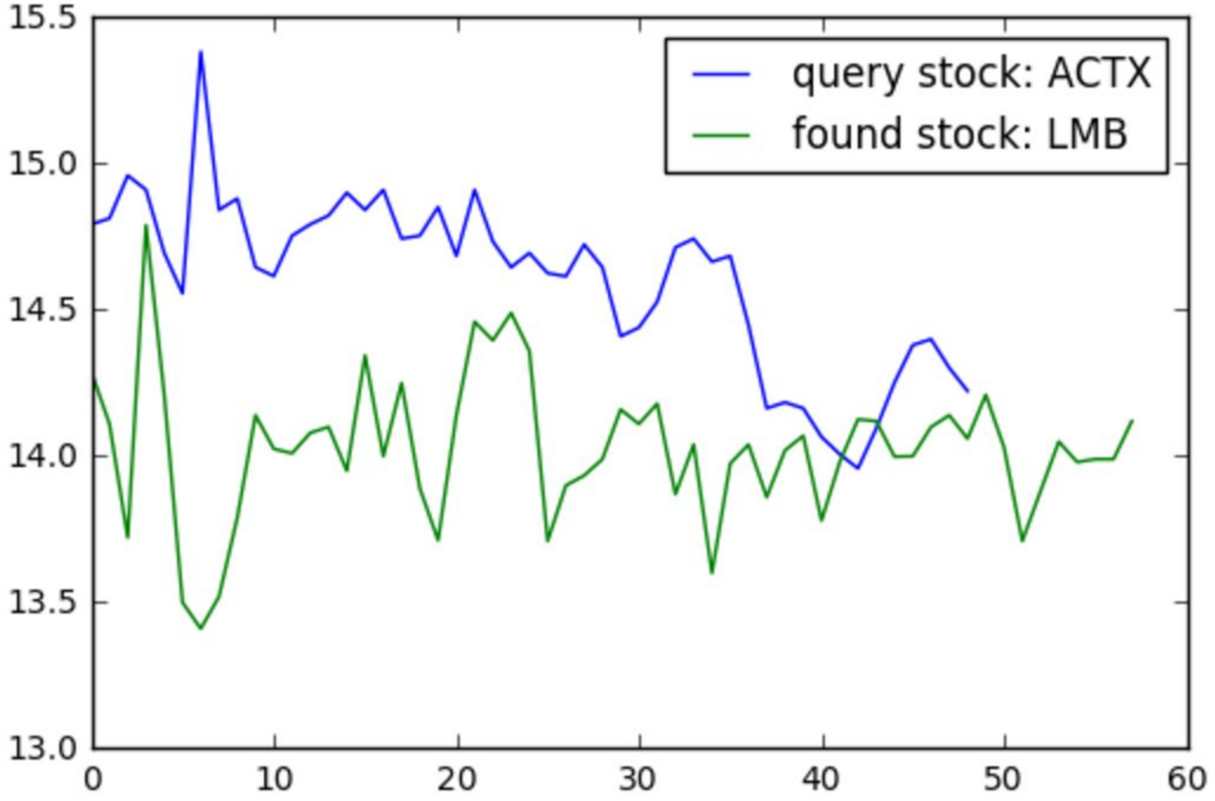
It's part of Glb X Guru Act Indx, NASDAQ: ACTX, historical close price from Apr 30th, 2015 to July 22nd, 2015

After executing the program, we find Limbach Hldgs Rg, NASDAQ: LMB has the minimum distance with the query. The distance is 

```
lmb      0.13145483870967742
```

(average)

Furthermore, the most similar range is Dec 7th, 2016 to Mar 14th, 2017
14.29,14.11,13.7216,14.79,14.21,13.5,13.41,13.52,13.79,14.14,14.025,14.01,14.08,14.1,13.95,14.3445,14,14.25,13.89,13.7125,14.14,14.46,14.396,14.49,14.36,13.71,13.9,13.934,13.99,14.16,14.11,14.18,13.87,14.04,13.6,13.975,14.04,13.86,14.02,14.07,13.78,13.98,14.1262,14.12,13.9985,14.0001,14.1,14.14,14.06,14.21,14.03,13.71,13.881,14.05,13.98,13.99,13.99,14.12

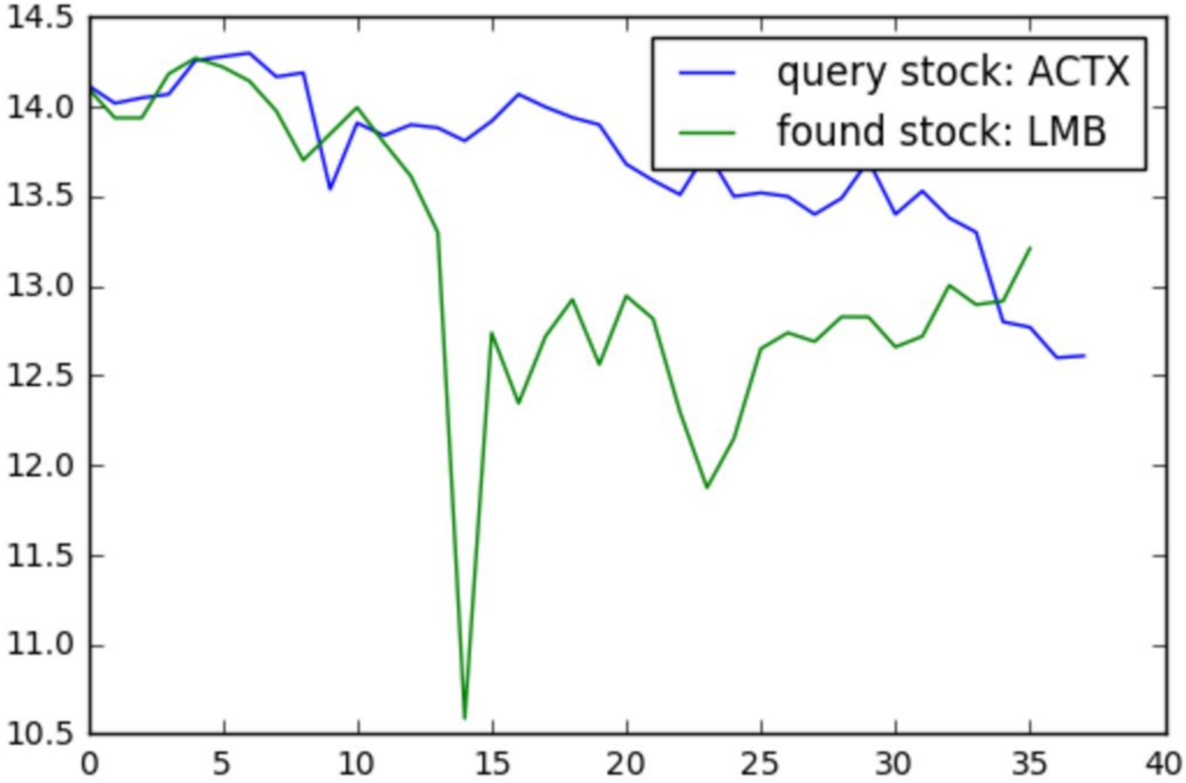


The above picture shows the comparison of two time stock

ACTX from Apr 30th, 2015 to July 22nd, 2015
LMB from Dec 7th, 2016 to Mar 14th, 2017

All execution log files (Hadoop log and path track data) are saved in experiment folder.

Conclusion



1. This picture shows the later trend of two stock. We can see they shows a similar drop trend. Which prove our program perform well in finding the most similar time series, and can be applied to predict the future trend of stock.
2. We are able to find out the distance of two stock time series using dynamic warping algorithm.
3. Hadoop MapReduce solved this problem in less than one minutes (you can refer log in the experiment file) for over 8,000 different stocks with over decades market history, which shows the MapReduce algorithm is very efficient.

Reference

1. Stock historical data download: https://stooq.com/db/h/
2. <Embedding-based Subsequence Matching in Time Series Databases>