Model to predict stock prices!

Presented by :
Aishwarya
Anil
Ankita

Outline

---

1. Problem Statement
2. Motivation
3. Our solution model
4. Machine learning algorithm
5. Implementation
6. Training Data
7. Results
8. Improvements
9. References
10. Questions

The Problem statement!

We are given 10 stocks with attributes.

We have to use ~2year’s stock data.

Build your MODEL, use the data, maximize the returns!

Motivation (For you!)

Company	26th Feb	3rd March	Rise	Allotted	Profit
Nilkamal	930	1064	134	38%	54,752
Aurobindo Pharma	588	698	110	15%	28,061
TCS	2210	2373	163	14%	10,325
SBI	155	183	28	9.5%	17,161

Our Solution Model

- Feature Vector (attributes) + The Close Price = Linear Regression.
- But stock prediction is that simple?
- Temporal Correlation!

  (for example, stock behavior one year ago is different from stock behavior now)
- We modify the linear regression model by allowing the vector [θ] to change with time.
- Its virtual money! (Still we care)

cont…

Do not put all your eggs in one basket!

--Mr Warren Buffett

Implementation of Above quote!

```
Change_predicted = ( [(AvgFeature) * (theta)] - ValueOn26thFeb )

If LOSS:
          if(less than 40)
                Value = 1000 / (abs(Change_predicted))
          else
                abs(change_predicted)
If GAIN:
          Value = (5)* Change_predicted
```

Linear Regression

- Linear regression is a type of supervised learning.
- In machine learning it is used for problems which involve set of input and output data and we want to predict output for future inputs.
- Linear regression algorithms learns from given input and output sets. So more the data better will be the prediction.
- Output takes continuous values.

Why linear regression to solve this problem ?

- A stocks can have multiple attributes e.g. opening price, closing price, high value, low value, volumes etc.
- Problem requires to predict whether stock value will go up or down.
- We are interested in closing price value and want to see whether it will go up or down.
- Closing price can have continuous values.
- Therefore linear regression.

How the system will learn ?

- We have defined a hypothesis function which is an n-dimensional linear expression.
- n is the number of attributes we are considering.
- A 1-dimensional hypothesis function will be like this $h_\theta(x)=\theta_0+\theta_1x$
- For our hypothesis function we will have an n+1 dimensional vector with some initial arbitrary value.
- We have also defined a cost function which will compute the difference between output by our hypothesis function and actual output.
- So our goal is to come up with a theta vector which will minimize the cost.

Gradient descent

- We have to find theta vector for which we should have minimum cost function output.
- Initial theta vector corresponds to a point on the cost function curve.
- So our problem is to find the local minima on that curve.
- Basic intuition behind Gradient descent is that we will take small steps towards local minima.
- We use partial differential equation to find the direction to move.
- For step size we use a constant value called learning rate

Contd...

- Choice of learning rate is very critical.
- Too small will slow down the learning rate
- Too big may lead to overshoot of local minima
- We have experimented with some values and come up with 0.01
- For how many iteration we will run gradient descent ?
- More the iteration more better will be the prediction.
- After some experimentation we finalized with 400 iteration.

Implementation

Used python to parsing data from file.

We have used octave for implementation of gradient descent algorithm.

Load data and separate into X(features) and Y(results)

--->Learn from last 10 days, use the theta for next 10 days, improve theta on these 10 days. Repeat!

Training Data

Feature List

```
Xi = [open_p,high_p,low_p,total_v,tradesNo,trunOver,Market Cap]

        X1     X2     X3    X4       X5       X6       X7
```

Target List

```
Yi = [close_p]
```

Contd..

Used 500 data sets from Feb 2014 to Feb 2015.

Example

CO_NAME	***Date***	***Open Price***	***Low Price***	***High Price***	***Total Volume***	***No. of Trades***	***Net Turnover -Rs. Thousand***	***Market Cap***	***Close Price***
Britannia Inds.	10/2/2014	905	919	903	62531	2455	56983.5	10917.85	**910.2**
		X1	X2	X3	X4	X5	X6	X7	**Y**

Results!!!

Prediction for next 10 days:

```
Britannia Inds        :        Fall 44 points
Vedanta               :        Fall 66 points
St Bk of India        :        Rise 150 points
Nilkamal Ltd          :        Rise 570 points
Aurobindo Ph.         :        Rise 236 points
TCS                   :        Rise 222 points
Jet Airways           :        Fall 54 points
Intrasoft Tech        :        Fall 97 points
ShRenukaSugar         :        Fall 60 points
Yes Bank              :        Rise 74 points
------------------------------------------------
Total                         Prediction, 1573 points
```

Distribution based on points!

Number of shares of each stock in a portfolio:

Britannia Inds	:	2.8%
Vedanta	:	4.2%
St Bk of India	:	9.5%
Nilkamal Ltd	:	36%
Aurobindo Ph.	:	15%
TCS	:	14%
Jet Airways	:	3.4%
Intrasoft Tech	:	6.0%
ShRenukaSugar	:	3.8%
Yes Bank	:	5.3%

Improvements

We got very less time to come up this solution,

- Use user data.
- Adding the user prediction using weighted majority.
- Expert weighted attributes.
- Risk Teller

Reference

1. https://en.wikipedia.org/wiki/Linear_regression
2. https://en.wikipedia.org/wiki/Gradient_descent
3. http://www.moneycontrol.com/
4. http://money.rediff.com/index.html
5. http://www.nseindia.com/

Questions?