Stock Prediction System

Renjie Zhang

December 18, 2017

1 Revision History

Date	Version	Notes
2017-09-25	1.0	create
2017-10-04	1.1	update

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

symbol	**unit**	**SI**
$	currency	dollar

2.2 Table of Symbols

symbol	**unit**	**description**
K	1	The Kernel use to solve the non-linear classification
y	integer	can be 1 or -1, use to represent the result. 1 means increase, -1 is decrease.
σ	percentage	Stock Price Volality (see 5.2.4 for detail)
C	price	stock daily price

2.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
Stock Prediction System	Stock Prediction System
T	Theoretical Model
SVM	Support Vector Machine
RBF	Radial basis function, a model of Support Vector Machine
K	Kernelling function
y	a level in a range of (-1 and 1)
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine
SVM	Support Vector Machine

Contents

3 Introduction

Stock price prediction is a popular and challenging topic nowadays. There were several prediction models such as linear statistical time series models. This project will introduce a stock prediction system is used to analyze the future trend of stocks. The prediction was provided by machine learning algorithms based on the historical data. The system will be run on a big data platform (Spark), in order to obtain the more accurate results. In this case, we need to setup a distributed system to support Spark. For more information please check the repository: https://github.com/renjiezhang/CAS-741

3.1 Purpose of Document

The purpose of this document is to explain how to implement a machine learning system for stock price prediction. With different algorithms and dataset, the system can be used for both short term and long term predictions. In this case, we will use Support Vector Machine to predict the future trend of stocks based on the daily historical data.

3.2 Scope of Requirements

The purpose of this software is to give the user a reference by calculating the possibility of the future price change. For example, it may go up with a chance of xx percent and go down with a chance of xx percent.

3.3 Characteristics of Intended Reader

Readers needs to have basic knowledge of the stock market and it is highly suggested to have some basic knowledge of big data, machine learning, especially Support Vector Machine. Readers with bachelor's degree of computer science or mathematic should have no problem to understand this document.

3.4 Organization of Document

This document will cover the configuration of the Spark distributed system for big data, the workflow of the program and the explanation of SVM algorithm. The template used to present the required information is based on SmithAndLai2005 and SmithEtAl2007

4 General System Description

The project displays the future trend of the stock prices for certain companies. Users are able to collect the data files of the companies and add them into the list. It loads the historical data provided by the user and calculate the future trend. A plot graph will be displayed as well. There will be two results come out, one is the to show the stock is going to increase or decrease in a certain period, other is the possibility of the result.

4.1 System Context

- User Responsibilities:
  - Prepare the historical data file. The files can be downloaded from yahoo finance https://finance.yahoo.com/ with a CSV format. The columns will be used is the date and closing price.
  - Decide the date time range of the historical period. The files can be downloaded by week, day and month.
  - Update the historical data set. User needs to copy the file into the folder ”dataset”, and do not change the file name.
- Stock Prediction System Responsibilities:
  - Load data from files and display errors when the loading failes
  - Display the plot of the stock
  - Predict the future trend based on the historical data with the possibilities

4.2 User Characteristics

The end user of Stock Prediction System should have some basic knowledge of machine learning algorithm (Support Vector Machine).

4.3 System Constraints

The system supports multiple operating systems. However, since it is running on Spark, it is more stable with Linux/Ubuntu. The program needs a distributed system consist of at least three computers, one will be the driver of Spark, the others will be the workers.

5 Specific System Description

This system is used to predict the future trend of stocks based on the historical data from Yahoo Finance using Support Vector Machine algorithm. The data from other resource is accepted if they fit the format.

5.1 Problem Description

Stock Prediction is a very important topic in financial industry. It is complicated because the stock price was affected by too many factors. One of the common prediction technology is to train and test the historical data using machine learning. Stock Prediction System is a tool to predict stocks based on machine learning and big data. It implements SVM classifier to train and test the historical data. With the development of Big Data and Machine learning, it is possible to obtain an more accurate prediction result. To delivery a more accurate result, a large scale of data is needed. In this case it is necessary to run the program on a distributed system platform such as Spark to accurate the calculation speed.

5.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Support Vector Machine : A supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis.
  More knowledge about SVM can be found at https://en.wikipedia.org/wiki/Support_vector_machine

- Spark : A big data platform that save and retrieve data from different machines in a format called RDD(Resilient Distributed Datasets). It provides a set of machine learning libraries and support different types of programming languages.
  More information about Spark can be found at https://spark.apache.org/

- RDD : Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.

- Distributed System : A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. It combines a set of computers to work on a single task as one machine.

- Training and Testing : A training set is a set of data used to discover potentially predictive relationships. A test set is a set of data used to assess the strength and utility of a predictive relationship. Test and training sets are used in intelligent systems, machine learning, genetic programming and statistics.

5.1.2 Physical System Description

The physical system of Stock Prediction System, is a distributed system consists by a driver and 2-3 works as shown in Figure 1

- Driver (Master) : The drive is the machine that send requests to the works and receive the response from, it sometimes called Master. Data will be converted into RDD format and equally assigned to each work. Drive itself does not do the actual works and there is only on drive in the distributed system.

- Worker (Slave): The workers are the machines which do the actual jobs and they are called slaves as well. They receive data from drive using RDD and return an RDD back to drive after certain processes. For each distributed system, there are at least 2 workers, otherwise it will work as a single node computers. Each worker has a copy of the program.

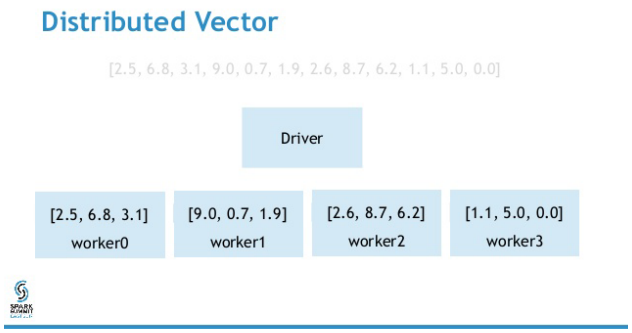


Figure 1

5.1.3 Goal Statements

- Prediction Result1: A result of the future stock price with its probability based on the input file. The result will be grouped by number of days of prediction period and companies.

5.2 Solution Characteristics Specification

The instance models that govern Stock Prediction System are presented in Subsection 5.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

5.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: Efficient Markets Hypothesis holds true. It posits that stock prices already reflected all available information and are therefore unpredictable.

A2: Independent identically distributed. A sequence or other collection of random variables is independent and identically distributed (i.i.d. or iid or IID) if each random variable has the same probability distribution as the others and all are mutually independent.

A3: The company is active on NASDAQ, because NASDAQ index will be used as cross reference to predict.

A4: The historical data is daily.

5.2.2 Theoretical Models

This section focuses on the general equations and laws that Stock Prediction System is based on.

Number	T1
Label	**Support Vector Machine**
Equation	$y = \beta_0 + \sum a_i y_i K(x(i), x)$
Description	The above equation gives a linear classification in a higher dimensional space and linearly classify in that space. x is an n-dimensional feature vector $x = (X_1, ...X_n)$. $y \in \{1, -1\}$ is the label, in a range of 1 and -1. This SVM replaces the inner product with a more general kernel function K which allows the input to be mapped to higher-dimensions. The range of i is the lengh of the price array. When feature x was calculated, it matches the length of the price array. In this case, RBF(Radial basis function) Kernel is used for price forecasting.
Source	https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
Ref. By	IM1

5.2.3 General Definitions

NA

5.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Stock Price Volatility**
Symbol	σ_s
SI Units	NA
Equation	$\frac{\sum_{i=t-n+1}^{t} \frac{C_i - C_{i-1}}{Ci-1}}{n}$
Description	Stock price is an average over the past n days of percent change in the given stocks price per day
Sources	https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
Ref. By	IM1

Number	DD2
Label	**Stock Momentum**
Symbol	NA
SI Units	NA
Equation	$\frac{\sum_{i=t-n+1}^{t} y}{n}$
Description	This is an average of the given stocks momentum over the past n days. Each day is labeled 1 if closing price that day is higher than the day before, and -1 [I had to modify the minus sign so that the correct symbol would show up in the generated pdf file. —SS] if the price is lower than the day before
Sources	https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
Ref. By	IM1

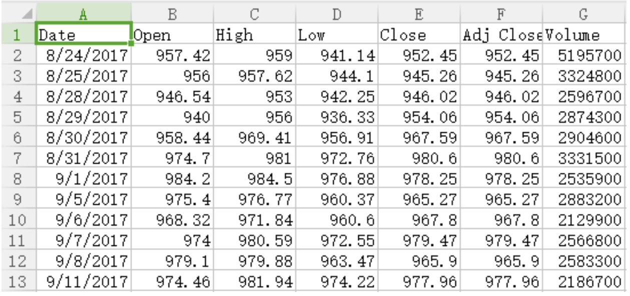
Number	DD3
Label	**Index Volatility**
Symbol	σ_i
SI Units	NA
Equation	$\frac{\sum_{i=t-n+1}^{t} \frac{I_i - I_{i-1}}{Ii-1}}{n}$
Description	This is an average over the past n days of percent change in the index's price perday
Sources	https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
Ref. By	IM1

Number	DD4
Label	**Index Momentum**
Symbol	NA
SI Units	NA
Equation	$\frac{\sum_{i=t-n+1}^{t} d}{n}$
Description	This is an average of the indexs momentum over the past n days. Each day is labeled 1 if closing price that day is higher than the day before, and 1 if the price is lower than the day before
Sources	https://www.cs.princeton.edu/sites/default/files/uploads/saahil_madge.pdf
Ref. By	IM1

5.2.5 Instance Models

This section transforms the problem defined in Section 5.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.2.4 to replace the abstract symbols in the models identified in Sections 5.2.2 and 5.2.3.

Number	IM1
Label	**Predict future trend by historical data**
Input	C_i the adjClose, the Date, the format of the input shown as the Figure 2
Output	1 or -1, the prediction result
Description	Date is the date of the trade, and adjClose stands for the adjusted closing price. It is a stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. Feature x and y in T1 will be calculated and a score which is also the probability will be predicted based on the x and y features. The result of 1 stands for increase and -1 means decrease.
Source	http://www.investopedia.com/terms/a/adjusted_closing_price.asp
Ref. By	



	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	8/24/2017	957.42	959	941.14	952.45	952.45	5195700
3	8/25/2017	956	957.62	944.1	945.26	945.26	3324800
4	8/28/2017	946.54	953	942.25	946.02	946.02	2596700
5	8/29/2017	940	956	936.33	954.06	954.06	2874300
6	8/30/2017	958.44	969.41	956.91	967.59	967.59	2904600
7	8/31/2017	974.7	981	972.76	980.6	980.6	3331500
8	9/1/2017	984.2	984.5	976.88	978.25	978.25	2535900
9	9/5/2017	975.4	976.77	960.37	965.27	965.27	2883200
10	9/6/2017	968.32	971.84	960.6	967.8	967.8	2129900
11	9/7/2017	974	980.59	972.55	979.47	979.47	2566800
12	9/8/2017	979.1	979.88	963.47	965.9	965.9	2583300
13	9/11/2017	974.46	981.94	974.22	977.96	977.96	2186700

Figure 2

5.2.6 Data Constraints

Tables 1 and 3 show the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
C	$C > 0$	$C > 0$	\$1000	NA

(*)

Table 2: Specification Parameter Values

Table 3: Output Variables

Var	Physical Constraints
y	1 or -1

5.2.7 Properties of a Correct Solution

A correct solution must exhibit. The solution is two results, 1 and -1, each result is with a percentage number. Any other outputs are incorrect. The result of 1 means the stock is increase, otherwise decreasing.

6 Requirements

The required functions are not complex in the system. It basically reads a data file and analysis the data then out put a result based on the analysis.

6.1 Functional Requirements

R1: The system must read the input data file provided by the user successfully. This is the first and every other functions rely on it. If the loading has any errors then the whole system will not work.

R2: As part of the interface and output, the system needs to generate a graph that explains the historical data. The graph is a plot consists by the points of stock price and date.

R3: The system needs to calculate the future trend of the stock by training and testing the historical data and finally obtained an acurte result. This is the most important function and it is also the core of the system.

R4: The input reading and data calculating must be valid and verified. Invalid data must be skipped and the data type and format must match

R5: There will be an result about the prediction. The result shows the stock will go up or go down with a probability in a percentage format. Reference to IM1

6.2 Nonfunctional Requirements

Performance is always a requirements for machine learning softwares. To increase the execution time, reduce the size of data and increase the number of workers will be good solutions.

7 Likely Changes

LC1: In short run, we may need intra-day/high frequency data to increase the accuracy.

LC2: Try to get more data with longer history and with a larger size of distributed system.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	T1	DD1	DD2	DD3	DD4	IM1	
T1							
DD1					X		
DD2						X	
DD3							
DD4							
IM1							

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	5.2.6	R1	R2	R3	R4	R5
IM1					X		
5.2.6			X	X	X		X
R1							X
R2							
R3							X
R4							X
R5							

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2	A3	A4
T1				
DD1				
DD2				
DD3				
DD4				
IM1	X	X	X	
LC1				X
LC2				

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

9 References

Modeling high-frequency limit order book dynamics with support vector machines PDF 2013
Predicting Stock Price Direction using Support Vector Machines PDF 2015

10 Appendix

NA

10.1 Symbolic Parameters

There is no Symbolic Parameters for this project.