CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

METHODS TO SOLVE ASSET BUBBLE IN FINANCE

A thesis submitted in partial fulfillment of the requirements
For the degree of Master of Science in
Applied Mathematics

by

Jaspreet Kaur

August 2013

The thesis of Jaspreet Kaur is approved:

Dr. Breen	Date
Dr. Penferov	Date
Dr. Jorge Balbás, Chair	Date

California State University, Northridge

Dedication

Jas' dedication

Acknowledgements

I would like to thank my advisor Dr. Jorge Balbas for his tremendous support and commitment towards my thesis research. I would like to special thank all of the faculty members at CSUN and my committee members Dr. Mary Rosen and Dr. Casaba Toth. Thank you so much for helping me to improve my work.

I would personally like to thank my husband, family and friends for supporting me through out my thesis work. With their support and encouragement, I was able to complete my thesis

Table of Contents

ABSTRACT

METHODS TO SOLVE ASSET BUBBLE IN FINANCE

By

Jaspreet Kaur

Master of Science in Applied Mathematics

Financial Market is very attracting topic in finance and mathematics world. Recently we have heard lot about Gold Prices inflations. It is the the hot topic in today's finance market. So how will be combine mathematics with today's asset changes? How can we determine the tale of asset's volatility for future? These are the questions which we will consider in this thesis. We will study non parametric estimator Floren Zmirou in local real time on compact domain with stochastic differential equation which has unknown drift and diffusion coeificents. Once we will have volatility from floren zmirou then we will able to use RKHS to estimates function which will extrapolate the tale of function.

0.0.1 Outline

As for the outline, i think you should have four chapters:

(1) Introduction:
(1.1) What problem are you trying to solve? Why is it important? What is your goal?
(1.2) Challenges
(1.3) Summary of how youÕve addressed those challenges, what have you found/built?
(1.4) Overview of following chapters

(2) Theoretical Background Ñ The Math:
(2.1) Where does your problem come from? (youÕll have to talk about stochastic PDEs, volatility, assets price modeling, martingales, etc.)
(2.2) Describe the math problem: parametric estimation, interpolation, extrapolation, minimization
(2.3) What is your solution? Based on all this theory connecting math and finance and assets and asset bubbles, how can you get an answer to your (math) problem as you described in chapter (1).

(3) Algorithm and Implementation Ñ The Software Tool:
Ñ We can discuss later, but you may want to consider different format options: do you want to have a section for each sub problem (acquiring data, fitting data, extrapolation, etc.) or you may want to group some of those and have a separate section talking about software tools that you have downloaded/modified, and tools you have developed
Ñ Also a section describing the Algorithm step by step.

(4) Numerical Examples Ñ Once your code works, run some experiments on real data, some that you know the answer, some that you donÕt know

You may want to have a final chapter with conclusions. future work, etc

Chapter 1

Implementation of an Asset Bubble Problem

1.1 Implementation

We used sage python to solve this problem. We orgnize our data and process in six classes. Each class is designed to store the information about Stock, Floren Zmirou Asset-Bubble, AssetBubbleDetection, Approximation, and Run.

1.1.1 Stock Class

Methods of Stock Class We obtain stock data by two types of methods. First, we will get it through Google Finance and second, we will get it from CSV file.

1. `IsNumber()`: Input for this function is rowValue. This function determine if rows of stock data is a numerical string or not.

2. `GetGoogleData()`: Input for this function are Ticker,days and period. This function obtains data for any stock from Google finance.

Algorithm 1 GetGoogleData ()

Inputs: Ticker, days, period
Steps

if the length of the Ticker is less than equal to 3 **then**
    exchange it with New York Stock Exchange (NYSE).
else
    exchange it with Natinal Association of Securities Dealers (NASD)
end if
We initialize current time in integer.
We will open Google Finance link.
We initialize dataList to read each line from opened link.
We initialize tickerData to be the list of array of dataList.
We will put stockPrices in list.
We initialize minuteData.

for minuteData in tickerData **do**
    We initialize datum and put split minuteData in commas in datum.
    We will append datum row one in float and store it in stockPrice list.
return stockPrices

---

3. `GetStockPrices()`: Input for this function is csv filename. This function obtains stock prices which are stored in csv file by filename. It will read the yahoo API based minute to minute data.

Algorithm 2 GetStockPrices ()

Inputs: filename
Steps

We saved stock Prices downloaded from yahoo in csv filename.
We initialize cr which will open and read the csv filename.
We skip the header.
Define c1 to be the empty list.
for row in cr **do**
    if the second row is numberical string **then**
        we add second row to c1.
return c1

4. `__init__()`: Input for this function is keyward Usage.This function will analyzie wheather csv filename is used or Ticker parameter for Google finance.

Algorithm 3 –init–()

Inputs: Keyword Usage
Steps

if filename is in keyward Usage **then**
    We will obtain Stock Prices from GetStockPrices in list of filename Ticker is used in keyward Usage
    We will obtain Stock Prices from GetGoogleData in a list of Ticker
elseWe will give exception of bad parameters

1.1.2 Floren Zmirou Class

Process to solve Floren Zmirou Now we have Stock Prices from Stock class. We will obtain list of sigma values from list of Stock Prices.We will explain this class in following algorithms:

T = 60*n where T is the minute to minute time period [0,T].
Now we will derive hn, x-step_size, and x values which will be used in Floren Zmirou Estimator.

5. `(Derive_hn)`: Input for this function is Stock Prices which we obtained from Stock class. This function will derive hn which will be used in Floren Zmirou Estimator.

6. `(x_step_size)`: Input for this function is Stock Prices which we obtained from Stock class. This function will derive x_step_size which will be used to create step size to generate x.

7. `(Derive_x_values)`: Input for this function is Stock Prices which we obtained from Stock class. This function will derive x_values which will be used in Floren

Algorithm 4 Derive hn ()

Inputs: Stock Prices S
Steps

n = length of Stock Prices S.
hn = 1.0/n**(1.0/3.0)
return hn =0

Algorithm 5 x step size ()

Inputs: Stock Prices S
Steps

We obtained hn from Derive hn function.
Double hn = 2 **hn
Difference = max S-min S
x_hn = Difference * Double hn
Return x_hn

Zmirou Estimator. Now we have all the ingridents which will be usefull to solve flo-

Algorithm 6 Derive x Values

Inputs: Stock Prices S
Steps

We will use x_hn from function x step size.
halfh_n = x_hn /2.0
We will define x to be equal to empty list.
We will append x with min S + halfh_n
We initilaize ex to be first element of x's list.
while ex < max S **do**
    ex = ex+x_hn
end while
We will append ex into x's list
Return x

ren zmirou estimator. We will implement floren Zmirou on the following functions. Floren Zmirou has Sublocal time, Local time, volatility Estimator and Indicator function.

8. `(Sublocal_Time)`: Input for this function are T,S,x,n,hn Output for this function will be LT$^n(x) = (\frac{T}{2nhn}) \sum_{i=1}^{n} 1_{|S_t(i)-x)|<hn}$

9. `(Local_Time)`: Input for this function are T,S,x,n,hn. Output for this function will be $L_T^n(x) = (\frac{T}{2nhn}) \sum_{i=1}^{n} 1_{|S\_t(i)-x)|<hn} * n(S(t(i+1)) - S(t(i))^2$

Algorithm 7 Sublocal Time

Inputs: T=Time,S= Stock Prices,x= grid points,n= number of total Stock Prices,hn=Step Size

Steps

```
sum = 0.0
scalar = T/(2.0*n*hn)
for i in range of the length of Stock Prices do
    We initialize Sti to be the ith element of the list of Stock prices
    absoluteValue = abs(Sti-x)
     We initilize indicatorValue to pass through indicator function
    if  absoluteValue is less than hn then
        sum = sum+indicatorValue return scalar*sum
```

Algorithm 8 Local Time

Inputs: T=Time,S= Stock Prices,x= grid points,n= number of total Stock Prices,hn=Step Size

Steps

```
sum = 0.0
scalar = T/(2.0*n*hn)
for for i in range of the length of Stock prices - 1 do
    We initialize Sti to be the ith element of the list of Stock prices
    We initialize Stj to be the jth element of the list of Stock prices
    absoluteValue = abs(Sti-x)
    Difference = (Stj-Sti)**2
     We initilize indicatorValue to pass through indicator function
    if  absoluteValue is less than hn then
        sum = sum+indicatorValue*n*Difference return scalar*sum
```

10. `(Volatility_Estimator Sn(x))`: Volatility Estimator is $\sigma^2(x)$.Input for this function are T,S,x,n,hn. Output for this function will be Local Time/Sublocal Time. We have $\sigma^2(x)$ for Stock Prices. Now we want to see how many stock prices are in each grid point. If there are less than 0 or 1 percent of stock prices in grid then we will exclude that grid point from our calculation process.

Algorithm 9 Volatility Estimator

Inputs: T=Time,S= Stock Prices,x= grid points,n= number of total Stock Prices,hn=Step Size
Steps

```
We will initialize ratio to be Local Time / Sublocal Time.
Return ratio. =0
```

11. `(DoGridAnalysis)`:Input for this function are T,S,x,n,hn. Output for this function give us the list of usable grid points and for each usable grid point, the list of Stock prices.

Algorithm 10 DoGridAnalysis

Inputs: T=Time,S= Stock Prices,x= grid points,n= number of total Stock Prices,hn=Step Size,Y = Y percent of total data points.
Steps

```
We initialize x to be useableGridPoints.
Create a empty dictionary and call it d.
We initialize stockPriceCount to be the length of Stock Prices.
for grid Points in x do
    create a dictionary where the keys are the grid points and the value is a list for each grid point
    for stockPrice in S do
        if satisfying the condition if true then add x value to corresponding Si then
```